```
"central control                                               kan 10/12-1992

module cc0;
"########################################################################
declarations;
"########################################################################
cc0d device 'mach220a';

"input signals:
clk, !res       pin;

"input from vsb
!vsbreq         pin;                "request from vsb
 mrq            pin;                "data request from vsb.
 vsbread        pin;                "read from vsb.
jmpc1, jmpc0    pin;                "jump code from vsb

suba5, suba4    pin;                "address inputs used in word counter
suba3, suba2    pin;                "address inputs used in word counter

"output to vsb

"enable arbitration on vsb
en              pin istype 'reg, buffer';

"enable address phase on vsb
ena             pin istype 'reg, buffer';

"last data to vsb
ld              pin istype 'reg_t, buffer';

"baio master
mast            pin istype 'reg, buffer';

"start address phase on vsb from baio
start           pin istype 'reg, buffer';

"memory acknowledge to vsb
mack            pin istype 'reg, buffer';

"output to DRAM controller

"address latch enable.
dale            pin istype 'reg, buffer';

"chip select siganl
dcs             pin istype 'reg, buffer';

"write signal
dwr             pin istype 'reg, buffer';

"burst
dburst          pin istype 'reg, buffer';
burstdm         pin istype 'reg, buffer';
```

```
"data handshake
waitd           pin istype 'reg, buffer';

"input from DRAM controller
drdcen          pin;            "clock enable read data
drack           pin;            "write data acknowledge.
mr              pin;            "memory ready.

"input from cpu
!dramcs         pin;            "decoded signal access to memory
!vsbcs          pin;            "decoded signal access to vsb
!ethcs          pin;            "decoded signal. port signal to ethernet
!cacs           pin;            "decoded signal. channel attention
!subwr          pin;            "write signal from cpu
burstc          pin;            "burst/writenear from cpu

"output to cpu

"clock enable read data
crdcen          pin istype 'reg, buffer'; "tristate

"write data acknowledge
cack            pin istype 'reg, buffer'; "tristate

"input from ethernet controller.
reade           pin;            "read signal
ads             pin;            "address strobe
blast           pin;            "last data
hold            pin;            "hold request
lock            pin;            "locked cycles

"output to ethernet controller.

"hold acknowledge
holda           pin istype 'reg, buffer';

"ready signal
erdy            pin istype 'reg, buffer';"enable rdy

"burst ready
ebrdy           pin istype 'reg, buffer';"enable brdy

port            pin istype 'reg, buffer';"write special to ethernet cont.

ca              pin istype 'reg, buffer';"channel attention ethernet cont.

"output enable address register from CPU, AREG.

oeareg          pin istype 'reg, buffer';

"output enable data register from CPU, DREG. (CPU write data)

oedreg          pin istype 'reg, buffer';

"CPU read data is clocked in DREG on every clock: clk.
```

```
"latch open to VSBD: CPU write data and VSB read data.

dle                pin istype 'reg, buffer'; "name changed from dclk

"output enable address register from VSB, VSBA.
"only used during access from VSB to data memory.

oevsba             pin istype 'reg_d, invert';

"output enable data register from VSB, VSBD.
"used during write access from VSB to data memory and
"during read access from CPU to VSB.

oevsbd             pin istype 'reg_d, invert';

"latch open to VSBA. Used during CPU access to VSB.

vsbaclk            pin istype 'reg, buffer';

"word counter for vsb access to baio

wc3, wc2, wc1, wc0        node istype 'reg_t, buffer';

c,x,z = .C., .X., .Z.;

"########################################################################
"                           Description
"########################################################################

"The control structure concists of 5 state machines sharing the same
"state variables, nr, n2, n1, n0. The five state machines are:

"CI:    controls CPU access to various io registers
"CV:    controls CPU access to VSB bus.
"CM:    controls CPU access to data memory.
"E :    controls ethernet access data memory.
"V :    controls vsb access to memory.

"The five state machines are enabled by five flip flops with the same
"names as the machines, e.g. the machine controlling access to memory
"vsb bus is enabled by flip flop v. The signal v.fb is an input to the
"state machine.

"The flip flops are controlled by a state machine called cycle.

"The state machine will only jump to a CPU cycle from the wait state.
"After the CPU cycle is executed, the state machine may return to the wait
"state. If the request from the CPU is still activated, the state machine
"will execute the same cycle again. A state variable cyst, cycle start,
"is used to avoid this. The state machine will only start a cycle if
"a request from the CPU and cyst both are activated. cyst is cleared
"in the beginning of a cycle. cyst is set again, when all requests from
"the CPU are deactivated.

nr, n2, n1, n0  node istype 'reg, buffer'; "state variables
ci,cv,cm,e,v    node istype 'reg, buffer'; "enable flip flops
```

```
np              node istype 'reg, buffer'; "state variable
i1, i0          node istype 'reg, buffer'; "state variable
cyst, bc        node istype 'reg, buffer'; "state variable

"#######################################################################
equations;
"#######################################################################

nr.c = clk;
n2.c = clk;
n1.c = clk;
n0.c = clk;

ci.c = clk;
cv.c = clk;
cm.c = clk;
 e.c = clk;
 v.c = clk;

np.c = clk;

i1.c = clk;
i0.c = clk;

cyst.c = clk;
bc.c = clk;

"outputs

en.c    = clk;   "enable arbitration on vsb
en     := v.fb;

ena.c   = clk;  "enable address phase on vsb

mast.c  = clk;  "baio master to vsb
mast   := cv.fb;

start.c = clk;  "baio master start address phase on vsb.

mack.c  = clk;  "memory acknowledge to vsb PAL.

dale.c  = clk;  "address latch enable to DRAM controller.

dcs.c   = clk;  "chip select signal to DRAM controller.

waitd.c = clk;  "data handshake signal to DRAM controller.

dwr.c   = clk;  "write signal to DRAM controller.
dwr    := !nr.fb;

dburst.c = clk; "burst signal to DRAM controller.
burstdm.c = clk; "DMA burst signal to DRAM controller.

crdcen.c = clk; "clock enable read data to CPU.

cack.c   = clk; "write acknowledge to CPU.
```

```
holda.c = clk;  "hold acknowledge to ethernet controller.

port.c  = clk;  "write scecial data to ethernet controller.
ca.c    = clk;  "channel attention  to ethernet controller.

oeareg.c = clk; "output enable AREG, address regiser from CPU.

oedreg.c = clk; "output enable DREG, write data register from CPU.

oevsba.c = clk; "output enable VSB address to memory

oevsbd.c = clk; "output enable VSBD. Data from VSB

vsbaclk.c= clk; "open VSBA latch.

dle.c    = clk; "open VSBD latch. VSB read in memory. CPU write to VSB.

erdy.c = clk;   "enable ready to ethernet controller
ebrdy.c= clk;   "enable burst ready to ethernet controller

"clock equations for word counter and last data to vsb interface

wc3.c = clk;
wc2.c = clk;
wc1.c = clk;
wc0.c = clk;
 ld.c = clk;

"output enable for feed back signals to CPU.

crdcen.oe = oeareg.q  & !np; "clock enable read data to CPU.
cack.oe   = oeareg.q  & !np; "write acknowledge to CPU.

"#######################################################################
                  "state machine for cycles.
"#######################################################################
declarations;

cycle = [ ci, cv, cm, e, v ];

wait     =  ^b00000;
cpuio    =  ^b10000;
cpuvsb   =  ^b01000;
cpumem   =  ^b00100;
ethernet =  ^b00010;
vsbtomem =  ^b00001;

wa = ^b00000;
li = ^b10000;
lv = ^b01000;
lm = ^b00100;
le = ^b00010;
ls = ^b00001;

ina = [cyst,dramcs,vsbcs,cacs,ethcs,hold,lock,vsbreq,np,res];
```

```
state_diagram cycle;

state wait:
"priority order: cpu, ethernet, vsb.

"hold acknowledge to ethernet;
holda := !dramcs & !vsbcs & !cacs & !ethcs & hold & !res;

oeareg := (dramcs # vsbcs # cacs # ethcs) & !res;

oevsba.d=
!dramcs & !vsbcs & !cacs & !ethcs & !hold & vsbreq & !res;

"see also equations for output signals.


"                     cy m  vb cs et h  l  rv np r
        case (ina == [x, x, x, x, x, x, x, x, x, 1]): wait;
             (ina == [0, x, x, x, x, x, x, x, x, 0]): wait;
             (ina == [1, 0, 0, 0, 0, 0, x, 0, x, 0]): wait;
             (ina == [1, 1, x, x, x, x, x, x, x, 0]): cpumem;
             (ina == [1, 0, 1, x, x, x, x, x, x, 0]): cpuvsb;
             (ina == [1, 0, 0, 1, x, x, x, x, x, 0]): cpuio;"ca,   ethernet
             (ina == [1, 0, 0, 0, 1, x, x, x, x, 0]): cpuio;"port, ethernet
             (ina == [1, 0, 0, 0, 0, 1, x, x, x, 0]): ethernet;
             (ina == [1, 0, 0, 0, 0, 0, x, 1, x, 0]): vsbtomem;
        endcase;

state cpuio:
"cpu access to io; priority order after cycle:
"       ethernet, vsb, cpu.

holda :=  hold & np & !res; "to ethernet

oevsba.d=        !hold &  vsbreq & np & !res;

"                     cy m  vb cs et h  l  rv np r
        case (ina == [x, x, x, x, x, x, x, x, x, 1]): wait;
             (ina == [x, x, x, x, x, x, x, x, 0, 0]): cpuio;
             (ina == [x, x, x, x, x, 1, x, x, 1, 0]): ethernet;
             (ina == [x, x, x, x, x, 0, x, 1, 1, 0]): vsbtomem;
             (ina == [x, x, x, x, x, 0, x, 0, 1, 0]): wait;
        endcase;


state cpuvsb:
"cpu access to vsb; priority order after cycle:
"       ethernet, vsb, cpu.

holda :=  hold & np & !res; "to ethernet

oevsba.d=        !hold &  vsbreq  & np & !res;

"                     cy m  vb cs et h  l  rv np r
        case (ina == [x, x, x, x, x, x, x, x, x, 1]): wait;
```

```
             (ina == [x, x, x, x, x, x, x, x, 0, 0]): cpuvsb;
             (ina == [x, x, x, x, x, 1, x, x, 1, 0]): ethernet;
             (ina == [x, x, x, x, x, 0, x, 1, 1, 0]): vsbtomem;
             (ina == [x, x, x, x, x, 0, x, 0, 1, 0]): wait;
        endcase;

state cpumem:
"cpu access to mem; priority order after cycle:
"       ethernet, vsb, cpu.

holda :=  hold & np & !res; "to ethernet

oevsba.d=        !hold &  vsbreq & np & !res;

"                     cy m  vb cs et h  l  rv np r
        case (ina == [x, x, x, x, x, x, x, x, x, 1]): wait;
             (ina == [x, x, x, x, x, x, x, x, 0, 0]): cpumem;
             (ina == [x, x, x, x, x, 1, x, x, 1, 0]): ethernet;
             (ina == [x, x, x, x, x, 0, x, 1, 1, 0]): vsbtomem;
             (ina == [x, x, x, x, x, 0, x, 0, 1, 0]): wait;
        endcase;

state ethernet:
"ethernet access to memory; priority order after cycle:
"       ethernet, vsb, cpu.

holda := holda & hold; "keep holda activated until hold = 0;

oevsba.d=        !hold & !lock &  vsbreq & np & !res;

"                        m  vb cs et h  l  rv np r
        case (ina == [x, x, x, x, x, x, x, x, x, 1]): wait;
             (ina == [x, x, x, x, x, x, x, x, 0, 0]): ethernet;
             (ina == [x, x, x, x, x, 1, x, x, 1, 0]): ethernet;
             (ina == [x, x, x, x, x, x, 1, x, 1, 0]): ethernet;
             (ina == [x, x, x, x, x, 0, 0, 1, 1, 0]): vsbtomem;
             (ina == [x, x, x, x, x, 0, 0, 0, 1, 0]): wait;
        endcase;

state vsbtomem:
"vsb access to memory; priority order after cycle:
"       cpu, ethernet, vsb.

holda :=  hold & np & !res &
          !dramcs &!vsbcs &!cacs &!ethcs;

"                        m  vb cs et h  l  rv np r
        case (ina == [x, x, x, x, x, x, x, x, x, 1]): wait;
             (ina == [x, x, x, x, x, x, x, x, 0, 0]): vsbtomem;
             (ina == [x, x, x, x, x, x, x, x, 1, 0]): wait;
        endcase;

"########################################################################
equations;
"########################################################################
```

```
"output enable AREG: hold until np

oeareg := oeareg & !np & !res;

"output enable DREG: hold until np

oedreg := oedreg & !np & !res;

"output enable VSBA: hold until np

oevsba.d= oevsba.q & !np & !res;

"cycle start, cyst := set # !reset & cyst.
"cyst is set when all request from the CPU are deactivated:
"set = !dramcs & !vsbcs & ! ethcs
"cyst is reset in the beginning of a cycle.
"bc := ci # cv # cm;
"reset = !bc & ( ci # cv # cm );

cyst := !dramcs & !vsbcs & !ethcs # res
      # cyst & !(!bc & (ci.fb # cv.fb # cm.fb ));

bc := ci.fb # cv.fb # cm.fb;


"########################################################################
declarations;
"########################################################################

t0  = ^b0000;
t1r = ^b1001;
t2r = ^b1011;
t3r = ^b1010;
t4r = ^b1110;
t5r = ^b1111;
t6r = ^b1101;
t7r = ^b1100;

t1w = ^b0001;
t2w = ^b0011;
t3w = ^b0010;
t4w = ^b0110;
t5w = ^b0111;
t6w = ^b0101;
t7w = ^b0100;

t0r = ^b1000;    " for reduction only


enner   =  [nr, n2, n1, n0];

in0     =  [mrq,vsbread,drdcen,drack,jmpc1,jmpc0,v.fb,mr,res] ;

"########################################################################
"              state_diagram for vsb access to memory
"########################################################################
```

```
state_diagram enner;

state t0r: "for reduction only

ena     := v.fb; "enable address phase on vsb
dale    :=!jmpc1 & jmpc0 & v.fb & !res; "address latch enable to DRAM
oevsbd.d=!jmpc1 & jmpc0 & !vsbread & v.fb & !res; "output enable write data
.


"                      mq rd ce ak j1 j0 v  mr r
        case (in0 == [x, x, x, x, x, x, x, x, 1]): t0;
             (in0 == [x, x, x, x, x, x, 0, x, x]): t0;
             (in0 == [x, x, x, x, 1, x, x, x, x]): t0;
             (in0 == [x, x, x, x, x, 0, x, x, x]): t0;
             (in0 == [x, 0, x, x, 0, 1, 1, x, 0]): t1w;
             (in0 == [x, 1, x, x, 0, 1, 1, x, 0]): t1r;
        endcase;

state t0:

ena     := v.fb; "enable address phase on vsb
dale    := !jmpc1 & jmpc0 & v.fb & !res; "address latch enable to DRAM
oevsbd.d= !jmpc1 & jmpc0 & !vsbread & v.fb & !res; "output enable write dat
a.

"equations for word counter. load counter and set ld.

wc0.t    =  wc0.q & !suba2 & !jmpc1 & jmpc0 & v.fb & !res
        # !wc0.q &  suba2 & !jmpc1 & jmpc0 & v.fb & !res;

wc1.t    =  wc1.q & !suba3 & !jmpc1 & jmpc0 & v.fb & !res
        # !wc1.q &  suba3 & !jmpc1 & jmpc0 & v.fb & !res;

wc2.t    =  wc2.q & !suba4 & !jmpc1 & jmpc0 & v.fb & !res
        # !wc2.q &  suba4 & !jmpc1 & jmpc0 & v.fb & !res;

wc3.t    =  wc3.q & !suba5 & !jmpc1 & jmpc0 & v.fb & !res
        # !wc3.q &  suba5 & !jmpc1 & jmpc0 & v.fb & !res;

ld.t     = !ld.q  &  (suba5 & suba4 & suba3 & suba2)
                             & !jmpc1 & jmpc0 & v.fb & !res
        #  ld.q  & !(suba5 & suba4 & suba3 & suba2)
                             & !jmpc1 & jmpc0 & v.fb & !res;


"                      mq rd ce ak j1 j0 v  mr r
        case (in0 == [x, x, x, x, x, x, x, x, 1]): t0;
             (in0 == [x, x, x, x, x, x, 0, x, x]): t0;
             (in0 == [x, x, x, x, 1, x, x, x, x]): t0;
             (in0 == [x, x, x, x, x, 0, x, x, x]): t0;
             (in0 == [x, 0, x, x, 0, 1, 1, x, 0]): t1w;
             (in0 == [x, 1, x, x, 0, 1, 1, x, 0]): t1r;
        endcase;

state t1r:
```

```
ena := v.fb; "enable address phase on vsb
dcs := v.fb; "chip select to DRAM controller.
burstdm := v.fb; "burstdma to DRAM controller.

"                      mq rd ce ak j1 j0 v  mr r
        case (in0 == [x, x, x, x, x, x, x, x, 1]): t0;
             (in0 == [x, x, x, x, x, x, 0, x, x]): t0;
             (in0 == [x, x, x, x, x, x, 1, x, 0]): t2r;
        endcase;

state t2r:

ena := v.fb; "enable address phase on vsb
dcs := v.fb; "chip select to DRAM controller.
burstdm := v.fb; "burstdma to DRAM controller.
waitd := mr; "wait to DRAM controller.

"                      mq rd ce ak j1 j0 v  mr r
        case (in0 == [x, x, x, x, x, x, x, x, 1]): t0;
             (in0 == [x, x, x, x, x, x, 0, x, x]): t0;
             (in0 == [x, x, x, x, x, x, 1, 0, 0]): t2r;
             (in0 == [x, x, x, x, x, x, 1, 1, 0]): t3r;
        endcase;

state t3r:

waitd := v.fb; "wait to DRAM controller.
mack  := v.fb & mrq & drdcen & !jmpc1 & jmpc0; "acknowledge to vsb pal
dle   := v.fb & mrq & drdcen & !jmpc1 & jmpc0; "open latch for data. VSBD

dcs   := v.fb & !jmpc1; "chip select to DRAM controller
burstdm := v.fb & !jmpc1; "burstdma to DRAM controller
ena   := v.fb & !jmpc1; "enable address phase on vsb
np    :=  jmpc1 & !jmpc0 & v.fb & !res; "cycle end

"equations for word counter. count counter and set ld

wc3.t  = v.fb & mrq & drdcen & !jmpc1 & jmpc0 & wc0.q & wc1.q & wc2.q;
wc2.t  = v.fb & mrq & drdcen & !jmpc1 & jmpc0 & wc0.q & wc1.q;
wc1.t  = v.fb & mrq & drdcen & !jmpc1 & jmpc0 & wc0.q;
wc0.t  = v.fb & mrq & drdcen & !jmpc1 & jmpc0;

ld.t   = v.fb & mrq & drdcen & !jmpc1 & jmpc0
       & !ld.q & wc0.q & wc1.q & wc2.q & wc3.q;

"                      mq rd ce ak j1 j0 v  mr r
        case (in0 == [x, x, x, x, x, x, x, x, 1]): t0;
             (in0 == [x, x, x, x, x, x, 0, x, x]): t0;
             (in0 == [x, x, 0, x, 0, 1, 1, x, 0]): t3r;
             (in0 == [0, x, x, x, 0, 1, 1, x, 0]): t3r;
             (in0 == [1, x, 1, x, 0, 1, 1, x, 0]): t4r;
             (in0 == [x, x, x, x, 1, 1, 1, x, 0]): t0 ; "new cycle
             (in0 == [x, x, x, x, 1, 0, 1, x, 0]): t7w; "end
        endcase;

state t4r:
```

```
ena := v.fb; "enable address phase on vsb
dcs := v.fb; "chip select to DRAM controller.
burstdm := v.fb; "burstdma to DRAM controller.

"                     mq rd ce ak j1 j0 v  mr r
        case (in0 == [x, x, x, x, x, x, x, x, 1]): t0;
             (in0 == [x, x, x, x, x, x, 0, x, x]): t0;
             (in0 == [x, x, x, x, x, x, 1, x, 0]): t3r;
        endcase;


state t1w:

ena    := v.fb; "enable address phase on vsb
dcs    := v.fb; "chip select to DRAM controller.
burstdm := v.fb; "burstdma to DRAM controller.
oevsbd.d= v.fb; "output enable write data.
waitd  := v.fb; "wait to DRAM controller.

"                     mq rd ce ak j1 j0 v  mr r
        case (in0 == [x, x, x, x, x, x, x, x, 1]): t0;
             (in0 == [x, x, x, x, x, x, x, x, 0]): t2w;
        endcase;

state t2w:

ena    := v.fb; "enable address phase on vsb
dcs    := v.fb; "chip select to DRAM controller.
burstdm := v.fb; "burstdma to DRAM controller.
oevsbd.d= v.fb; "output enable write data.
waitd  := v.fb; "wait to DRAM controller.

"                     mq rd ce ak j1 j0 v  mr r
        case (in0 == [x, x, x, x, x, x, x, x, 1]): t0;
             (in0 == [x, x, x, x, x, x, 0, x, x]): t0;
             (in0 == [x, x, x, x, x, x, 1, 0, 0]): t2w;
             (in0 == [x, x, x, x, x, x, 1, 1, 0]): t3w;
        endcase;

state t3w:

ena    := !jmpc1 &          v.fb;         "enable address phase on vsb
burstdm := !jmpc1 &         v.fb;         ; "burstdma to DRAM controller
oevsbd.d= !jmpc1          & v.fb & !res; "output enable write data.
np     :=  jmpc1 & !jmpc0 & v.fb & !res; "cycle end
waitd  := !(mrq & drack); "wait to DRAM controller
mack   :=  (mrq & drack);

"equations for word counter. count counter and set ld

wc3.t  = v.fb & mrq & drack  & !jmpc1 & jmpc0 & wc0.q & wc1.q & wc2.q;
wc2.t  = v.fb & mrq & drack  & !jmpc1 & jmpc0 & wc0.q & wc1.q;
wc1.t  = v.fb & mrq & drack  & !jmpc1 & jmpc0 & wc0.q;
wc0.t  = v.fb & mrq & drack  & !jmpc1 & jmpc0;
```

```
ld.t   = v.fb & mrq & drack  & !jmpc1 & jmpc0
       & !ld.q & wc0.q & wc1.q & wc2.q & wc3.q;

"                     mq rd ce ak j1 j0 v  mr r
        case (in0 == [x, x, x, x, x, x, x, x, 1]): t0;
             (in0 == [x, x, x, x, x, x, 0, x, x]): t0;
             (in0 == [x, x, x, 0, 0, 1, 1, x, 0]): t3w;
             (in0 == [0, x, x, x, 0, 1, 1, x, 0]): t3w;
             (in0 == [1, x, x, 1, 0, 1, 1, x, 0]): t4w;
             (in0 == [x, x, x, x, 1, 1, 1, x, 0]): t0 ; "new cycle
             (in0 == [x, x, x, x, 1, 0, 1, x, 0]): t7w; "end
        endcase;

state t4w:
ena    := v.fb; "enable address phase on vsb
dcs    := v.fb; "chip select to DRAM controller.
burstdm := v.fb; "burstdma to DRAM controller.
oevsbd.d= v.fb; "output enable write data.
waitd  := v.fb; "wait to DRAM controller

"                     mq rd ce ak j1 j0 v  mr r
        case (in0 == [x, x, x, x, x, x, x, x, 1]): t0;
             (in0 == [x, x, x, x, x, x, 0, x, x]): t0;
             (in0 == [x, x, x, x, x, x, 1, x, 0]): t3w;
        endcase;

state t7w:
"                     mq rd ce ak j1 j0 v  mr r
        case (in0 == [x, x, x, x, x, x, x, x, 1]): t0;
             (in0 == [x, x, x, x, x, x, 0, x, x]): t0;
             (in0 == [x, x, x, x, x, x, 1, x, 0]): t0;
        endcase;

"#######################################################################
"         state diagram for CPU access to VSB bus.
"#######################################################################

declarations;

in1  =  [ vsbcs, subwr, jmpc0, cv.fb, res ];

state_diagram enner;

state t0:

oedreg := vsbcs &  subwr & cv.fb & !res; "output enable data from CPU.
dle    := vsbcs &  subwr & cv.fb & !res; "latch write data in VSBD.
oevsbd.d= vsbcs & !subwr & cv.fb & !res; "output enable data from VSB.
vsbaclk:= vsbcs & cv.fb & !res; "open VSBA latch;

"                     rq wr e  cv r
        case (in1 == [x, x, x, x, 1]): t0;
             (in1 == [x, x, x, 0, x]): t0;
             (in1 == [0, x, x, x, x]): t0;
             (in1 == [1, 0, x, 1, 0]): t1r; "read cycle on VSB
             (in1 == [1, 1, x, 1, 0]): t1w; "write cycle on VSB
        endcase;

state t1r:

start  := cv.fb; "start address handshake on vsb.
oevsbd.d= cv.fb; "output enable data from VSB.

"                     rq wr e  cv r
        case (in1 == [x, x, x, x, 1]): t0;
             (in1 == [x, x, x, x, 0]): t2r; "read cycle on VSB
        endcase;

state t2r:

start  := cv.fb; "start address handshake on vsb.
oevsbd.d= cv.fb; "output enable data from VSB.
np     := jmpc0 & cv.fb; "cycle end

"                     rq wr e  cv r
        case (in1 == [x, x, x, x, 1]): t0;
             (in1 == [x, x, x, 0, x]): t0;
             (in1 == [x, x, 0, 1, 0]): t2r; "wait for VSB
             (in1 == [x, x, 1, 1, 0]): t3r; "cycle end
        endcase;

state t3r:

"read data is clocked in DREG on clk.
crdcen := cv.fb; "clock enable read data to CPU.
oevsbd.d= cv.fb; "output enable data from VSB.

"                     rq wr e  cv r
        case (in1 == [x, x, x, x, 1]): t0;
             (in1 == [x, x, x, 0, x]): t0;
             (in1 == [x, x, x, 1, 0]): t0;
        endcase;

state t1w:

start := cv.fb; "start address handshake on vsb.

"                     rq wr e  cv r
        case (in1 == [x, x, x, x, 1]): t0;
             (in1 == [x, x, x, x, 0]): t2w; "write cycle on VSB
        endcase;

state t2w:

start := cv.fb; "start address handshake on vsb.
np     := jmpc0 & cv.fb; "cycle end

"                     rq wr e  cv r
        case (in1 == [x, x, x, x, 1]): t0;
             (in1 == [x, x, x, 0, x]): t0;
             (in1 == [x, x, 0, 1, 0]): t2w; "wait for VSB
             (in1 == [x, x, 1, 1, 0]): t3w; "cycle end
        endcase;

state t3w:
"                     rq wr e  cv r
        case (in1 == [x, x, x, x, 1]): t0;
             (in1 == [x, x, x, 0, x]): t0;
             (in1 == [x, x, x, 1, 0]): t0;
        endcase;

"#######################################################################
"            State diagram for ethernet access to memory.
"#######################################################################

declarations;

in2 = [ ads, reade, blast, drdcen, drack, e.fb, res ];

state_diagram enner;

state t0:

dale   := ads & e.fb & !res; "address latch enable to DRAM controller.

"                     ad rd l ce ak  e  r
        case (in2 == [x, x, x, x, x, x, 1]): t0;
             (in2 == [x, x, x, x, x, 0, x]): t0;
             (in2 == [0, x, x, x, x, x, x]): t0;
             (in2 == [1, 0, x, x, x, 1, 0]): t1w;
             (in2 == [1, 1, x, x, x, 1, 0]): t1r;
        endcase;

"read cycle.

state t1r:

dcs    := e.fb; "chip select to DRAM controller.
burstdm:= e.fb;  "burst DMA signal to DRAM controller.

"                     ad rd l ce ak  e  r
        case (in2 == [x, x, x, x, x, x, 1]): t0;
             (in2 == [x, x, x, x, x, x, 0]): t2r;
        endcase;

state t2r:

dcs    := e.fb; "chip select to DRAM controller.
burstdm:= e.fb;  "burst DMA signal to DRAM controller.
erdy   := e.fb &  blast; "enable ready to ethernet controller.
ebrdy  := e.fb & !blast; "eanble burst ready to ethernet controller.

"                     ad rd l ce ak  e  r
        case (in2 == [x, x, x, x, x, x, 1]): t0;
             (in2 == [x, x, x, x, x, 0, x]): t0;
             (in2 == [x, x, 0, x, x, 1, 0]): t3r;
             (in2 == [x, x, 1, x, x, 1, 0]): t6r;
        endcase;
```

```
state t3r:

dcs     := e.fb; "chip select to DRAM controller.
burstdm:= e.fb;  "burst DMA signal to DRAM controller.
ebrdy   := e.fb; "enable burst ready to ethernet controller.

"                     ad rd l ce ak  e  r
        case (in2 == [x, x, x, x, x, x, 1]): t0;
             (in2 == [x, x, x, x, x, 0, x]): t0;
             (in2 == [x, x, x, 0, x, 1, 0]): t3r; "wait for word 1
             (in2 == [x, x, x, 1, x, 1, 0]): t4r;
        endcase;

state t4r:

dcs     := e.fb; "chip select to DRAM controller.
burstdm:= e.fb;  "burst DMA signal to DRAM controller.
ebrdy   := e.fb; "enable burst ready to ethernet controller.

"                     ad rd l ce ak  e  r
        case (in2 == [x, x, x, x, x, x, 1]): t0;
             (in2 == [x, x, x, x, x, 0, x]): t0;
             (in2 == [x, x, x, 0, x, 1, 0]): t4r; "wait for word 2
             (in2 == [x, x, x, 1, x, 1, 0]): t5r;
        endcase;

state t5r:

dcs     := e.fb; "chip select to DRAM controller.
burstdm:= e.fb;  "burst DMA signal to DRAM controller.
ebrdy   := e.fb & !drdcen; "enable burst ready to ethernet controller.
erdy    := e.fb  &  drdcen; "enable data ready to ethernet controller.

"                     ad rd l ce ak  e  r
        case (in2 == [x, x, x, x, x, x, 1]): t0;
             (in2 == [x, x, x, x, x, 0, x]): t0;
             (in2 == [x, x, x, 0, x, 1, 0]): t5r; "wait for word 3
             (in2 == [x, x, x, 1, x, 1, 0]): t6r;
        endcase;

state t6r:

dcs     := e.fb & !drdcen; "chip select to DRAM controller.
burstdm:= e.fb & !drdcen;  "burst DMA signal to DRAM controller.
erdy    := e.fb; "enable data ready to ethernet controller.
np      := e.fb & drdcen; "cycle end

"                     ad rd l ce ak  e  r
        case (in2 == [x, x, x, x, x, x, 1]): t0;
             (in2 == [x, x, x, x, x, 0, x]): t0;
             (in2 == [x, x, x, 0, x, 1, 0]): t6r; "wait for last word
             (in2 == [x, x, x, 1, x, 1, 0]): t0;
        endcase;

"write cycle
```

```
state t1w:

dcs    := e.fb; "chip select to DRAM controller.
burstdm:= e.fb;  "burst DMA signal to DRAM controller.

"                     ad rd l ce ak  e  r
        case (in2 == [x, x, x, x, x, x, 1]): t0;
             (in2 == [x, x, x, x, x, x, 0]): t2w;
        endcase;

state t2w:

dcs    := e.fb; "chip select to DRAM controller.
burstdm:= e.fb;  "burst DMA signal to DRAM controller.
erdy   := e.fb &  blast; "enable ready to ethernet controller.
ebrdy  := e.fb & !blast; "eanble burst ready to ethernet controller.

"                     ad rd l ce ak  e  r
        case (in2 == [x, x, x, x, x, x, 1]): t0;
             (in2 == [x, x, x, x, x, 0, x]): t0;
             (in2 == [x, x, 0, x, x, 1, 0]): t3w;
             (in2 == [x, x, 1, x, x, 1, 0]): t6w;
        endcase;

state t3w:

dcs    := e.fb; "chip select to DRAM controller.
burstdm:= e.fb;  "burst DMA signal to DRAM controller.
ebrdy  := e.fb; "enabel burst ready to ethernet controller.

"                     ad rd l ce ak  e  r
        case (in2 == [x, x, x, x, x, x, 1]): t0;
             (in2 == [x, x, x, x, x, 0, x]): t0;
             (in2 == [x, x, x, x, 0, 1, 0]): t3w; "write word 1
             (in2 == [x, x, x, x, 1, 1, 0]): t4w;
        endcase;

state t4w:

dcs    := e.fb; "chip select to DRAM controller.
burstdm:= e.fb;  "burst DMA signal to DRAM controller.
ebrdy  := e.fb; "enabel burst ready to ethernet controller.

"                     ad rd l ce ak  e  r
        case (in2 == [x, x, x, x, x, x, 1]): t0;
             (in2 == [x, x, x, x, x, 0, x]): t0;
             (in2 == [x, x, x, x, 0, 1, 0]): t4w; "write word 2
             (in2 == [x, x, x, x, 1, 1, 0]): t5w;
        endcase;

state t5w:

dcs    := e.fb; "chip select to DRAM controller.
burstdm:= e.fb;  "burst DMA signal to DRAM controller.
ebrdy  := e.fb & !drack; "enabel burst ready to ethernet controller.
```

```
 erdy  := e.fb &  drack; "enabel data ready to ethernet controller.

"                      ad rd l ce ak  e  r
        case (in2 == [x, x, x, x, x, x, 1]): t0;
             (in2 == [x, x, x, x, x, 0, x]): t0;
             (in2 == [x, x, x, x, 0, 1, 0]): t5w; "write word 3
             (in2 == [x, x, x, x, 1, 1, 0]): t6w;
        endcase;

state t6w:

dcs    := e.fb & !drack; "chip select to DRAM controller.
burstdm:= e.fb & !drack;  "burst DMA signal to DRAM controller.
erdy   := e.fb;  "enable data ready to ethernet controller.
np     := e.fb &  drack; "cycle end

"                      ad rd l ce ak  e  r
        case (in2 == [x, x, x, x, x, x, 1]): t0;
             (in2 == [x, x, x, x, x, 0, x]): t0;
             (in2 == [x, x, x, x, 0, 1, 0]): t6w; "write last word
             (in2 == [x, x, x, x, 1, 1, 0]): t0;
        endcase;

"############################################################################
"                      CPU access to memory.
"############################################################################

declarations;

in3 = [ dramcs, subwr, burstc, drdcen, drack, cm.fb, res ];

"read data is clocked in DREG on every clk.

state_diagram enner;

state t0:

dale := dramcs & cm.fb & !res; "address latch enable to DRAM cont.

"                      rq wr b ce ak cm  r
        case (in3 == [x, x, x, x, x, x, 1]): t0;
             (in3 == [x, x, x, x, x, 0, x]): t0;
             (in3 == [0, x, x, x, x, x, x]): t0;
             (in3 == [1, 1, x, x, x, 1, 0]): t1w;
             (in3 == [1, 0, x, x, x, 1, 0]): t1r;
        endcase;

"read cycle.

state t1r:

dcs    := cm.fb; "chip select to DRAM controller.
dburst := cm.fb & burstc; "burst signal to DRAM controller.

"                      rq wr b ce ak cm  r
        case (in3 == [x, x, x, x, x, x, 1]): t0;
```

```
             (in3 == [x, x, x, x, x, x, 0]): t2r;
        endcase;

state t2r:

dcs    := cm.fb; "chip select to DRAM controller.
dburst := cm.fb & dburst; "burst signal to DRAM controller.

"                     rq wr b ce ak cm  r
        case (in3 == [x, x, x, x, x, x, 1]): t0;
             (in3 == [x, x, x, x, x, 0, x]): t0;
             (in3 == [x, x, 1, x, x, 1, 0]): t3r;
             (in3 == [x, x, 0, x, x, 1, 0]): t6r;
        endcase;

state t3r:

dcs    := cm.fb; "chip select to DRAM controller.
dburst := cm.fb & dburst; "burst signal to DRAM controller.
crdcen := cm.fb & drdcen; "clock enable read data to CPU.

"                     rq wr b ce ak cm  r
        case (in3 == [x, x, x, x, x, x, 1]): t0;
             (in3 == [x, x, x, x, x, 0, x]): t0;
             (in3 == [x, x, x, 0, x, 1, 0]): t3r; "wait for word 1.
             (in3 == [x, x, x, 1, x, 1, 0]): t4r;
        endcase;

state t4r:

dcs    := cm.fb; "chip select to DRAM controller.
dburst := cm.fb & dburst; "burst signal to DRAM controller.
crdcen := cm.fb & drdcen; "clock enable read data to CPU.

"                     rq wr b ce ak cm  r
        case (in3 == [x, x, x, x, x, x, 1]): t0;
             (in3 == [x, x, x, x, x, 0, x]): t0;
             (in3 == [x, x, x, 0, x, 1, 0]): t4r; "wait for word 2.
             (in3 == [x, x, x, 1, x, 1, 0]): t5r;
        endcase;

state t5r:

dcs    := cm.fb; "chip select to DRAM controller.
dburst := cm.fb & dburst; "burst signal to DRAM controller.
crdcen := cm.fb & drdcen; "clock enable read data to CPU.

"                     rq wr b ce ak cm  r
        case (in3 == [x, x, x, x, x, x, 1]): t0;
             (in3 == [x, x, x, x, x, 0, x]): t0;
             (in3 == [x, x, x, 0, x, 1, 0]): t5r; "wait for word 3.
             (in3 == [x, x, x, 1, x, 1, 0]): t6r;
        endcase;

state t6r:
```

```
dcs    := cm.fb; "chip select to DRAM controller.
dburst := cm.fb & dburst; "burst signal to DRAM controller.
crdcen := cm.fb & drdcen; "clock enable read data to CPU.
np     := cm.fb & drdcen; "cycle end

"                      rq wr b ce ak cm  r
        case (in3 == [x, x, x, x, x, x, 1]): t0;
             (in3 == [x, x, x, x, x, 0, x]): t0;
             (in3 == [x, x, x, 0, x, 1, 0]): t6r; "wait for last word.
             (in3 == [x, x, x, 1, x, 1, 0]): t7w;
        endcase;

"write cycle.

state t1w:

dcs    := cm.fb; "chip select to DRAM controller.
cack   := cm.fb; "write acknowledge to CPU.
oedreg := cm.fb; "output enable write data from CPU

"                      rq wr b ce ak cm  r
        case (in3 == [x, x, x, x, x, x, 1]): t0;
             (in3 == [x, x, x, x, x, x, 0]): t2w; "
        endcase;

state t2w:

dcs    := cm.fb; "chip select to DRAM controller.
oedreg := cm.fb; "output enable write data from CPU

"                      rq wr b ce ak cm  r
        case (in3 == [x, x, x, x, x, x, 1]): t0;
             (in3 == [x, x, x, x, x, 0, x]): t0;
             (in3 == [x, x, x, x, x, 1, 0]): t3w; "
        endcase;

state t3w:

dcs    := cm.fb; "chip select to DRAM controller.
oedreg := cm.fb & !drack; "output enable write data from CPU
np     := cm.fb &  drack; "output enable write data from CPU

"                      rq wr b ce ak cm  r
        case (in3 == [x, x, x, x, x, x, 1]): t0;
             (in3 == [x, x, x, x, x, 0, x]): t0;
             (in3 == [x, x, x, x, 0, 1, 0]): t3w; "write one word.
             (in3 == [x, x, x, x, 1, 1, 0]): t7w;
        endcase;

state t7w:
"                      rq wr b ce ak cm  r
        case (in3 == [x, x, x, x, x, x, 1]): t0;
             (in3 == [x, x, x, x, x, 0, x]): t0;
             (in3 == [x, x, x, x, x, 1, 0]): t0; "
        endcase;
```

```
"#######################################################################
"                          CPU I/O cycles
"#######################################################################
"generates channel attention to ethernet controller
"generates port signal to ethernet controller
"reads error status from PAL par. read from channel attention address

declarations;

s0 = ^b00;
s1 = ^b01;
s2 = ^b11;
s3 = ^b10;

in4 = [cacs, ethcs, ci.fb, res ];

state_diagram [i1,i0];

state s0:
ca      := cacs  &   ci.fb & subwr; "channel attention, ethernet.
oedreg  := ethcs &   ci.fb; "output enable port data


"                     ca po ci r
        case (in4 == [x, x, x, 1]): s0;
             (in4 == [x, x, 0, x]): s0;
             (in4 == [0, 0, 1, 0]): s0; "
             (in4 == [1, x, 1, 0]): s1; "
             (in4 == [0, 1, 1, 0]): s1; "
        endcase;

state s1:
ca     := cacs  & ci.fb & subwr; "channel attention, ethernet
oedreg := ethcs & ci.fb; "output enable port data
port   := ethcs & ci.fb; "port signal to ethernet controller

"                     ca po ci r
        case (in4 == [x, x, x, 1]): s0;
             (in4 == [x, x, 0, x]): s0;
             (in4 == [0, 0, 1, 0]): s0; "
             (in4 == [1, x, 1, 0]): s2; "
             (in4 == [0, 1, 1, 0]): s2; "
        endcase;

state s2:
ca     := cacs  & ci.fb & subwr; "channel attention to ethernet controller
oedreg := ethcs & ci.fb; "output enable port data
port   := ethcs & ci.fb; "port signal to ethernet controller
cack   :=         ci.fb; "acknowledge to CPU.
np     :=         ci.fb;
crdcen := cacs  & ci.fb & !subwr; "read errors

"                     ca po ci r
        case (in4 == [x, x, x, 1]): s0;
             (in4 == [x, x, 0, x]): s0;
             (in4 == [0, 0, 1, 0]): s0; "
             (in4 == [1, x, 1, 0]): s3; "
             (in4 == [0, 1, 1, 0]): s3; "
        endcase;


state s3:
ca     := cacs  & ci.fb & subwr; "channel attention to ethernet controller
oedreg := ethcs & ci.fb; "output enable port data

"                     ca po ci r
        case (in4 == [x, x, x, 1]): s0;
             (in4 == [x, x, 0, x]): s0;
             (in4 == [x, x, 1, 0]): s0; "
        endcase;


"#######################################################################
"               test vectors for vsb access to memory
"#######################################################################


test_vectors
([clk,dramcs,vsbcs,cacs,ethcs,dramcs ,
hold,vsbreq,mrq,vsbread,jmpc1,jmpc0,drdcen,drack,res]
 -> [cycle,cyst,np,enner,oeareg,oedreg,oevsba,oevsbd,dale,dcs,
     dwr,dwr,en,ena,dle]);
"           d       v
"   d       r   v v s     d                   o o o o
"   r v   e a   s s b j j r d      c       e  e e e e
"   a s c t m h b b r m m d r      y  c    n  a d v v d           d
" c m b a h s o r a e p p c a r     c  y    n  r r s s a d d d   e c
" l c c c c e l e l a c c e c e     l  s n  e  e e b b l c r w e n l
" k s s s s l h q e d 1 0 n k s     e  t p  r  g g a d e s d r n a k
 [c,x,x,x,x,x,x,x,x,x,x,x,0,0,1]->[wa,1,0,t0 ,0,0,0,0,0,0,0,1,0,0,0];
 [c,0,0,0,0,0,0,0,x,x,x,x,0,0,0]->[wa,1,0,t0 ,0,0,0,0,0,0,0,1,0,0,0];
 [c,0,0,0,0,0,0,1,x,x,x,x,0,0,0]->[ls,1,0,t0 ,0,0,1,0,0,0,0,1,1,0,0];
 [c,x,x,x,x,x,x,x,0,x,x,x,0,0,0]->[ls,x,0,t0 ,0,0,1,0,0,0,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,1,1,x,x,0,0,0]->[ls,x,0,t1r,0,0,1,0,1,0,1,0,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,x,x,0]->[ls,x,0,t2r,0,0,1,0,0,1,1,0,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,x,x,0]->[ls,x,0,t3r,0,0,1,0,0,1,1,0,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,0,x,0]->[ls,x,0,t3r,0,0,1,0,0,1,1,0,1,1,0];
 [0,x,x,x,x,x,x,x,x,x,x,x,1,x,0]->[ls,x,0,t3r,0,0,1,0,0,1,1,0,1,1,1];
 [c,x,x,x,x,x,x,x,x,x,x,x,1,x,0]->[ls,x,0,t4r,0,0,1,0,0,1,1,0,1,1,1];"10
 [c,x,x,x,x,x,x,x,x,x,0,0,x,x,0]->[ls,x,0,t4r,0,0,1,0,0,0,1,0,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,0,1,x,x,0]->[ls,x,0,t1r,0,0,1,0,1,0,1,0,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,x,x,0]->[ls,x,0,t2r,0,0,1,0,0,1,1,0,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,x,x,0]->[ls,x,0,t3r,0,0,1,0,0,1,1,0,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,0,x,0]->[ls,x,0,t3r,0,0,1,0,0,1,1,0,1,1,0];
 [0,x,x,x,x,x,x,x,x,x,x,x,1,x,0]->[ls,x,0,t3r,0,0,1,0,0,1,1,0,1,1,1];
 [c,x,x,x,x,x,x,x,x,x,x,x,1,x,0]->[ls,x,0,t4r,0,0,1,0,0,1,1,0,1,1,1];"17
"           d       v
"   d       r   v v s     d                   o o o o
"   r v   e a   s s b j j r d      c       e  e e e e
"   a s c t m h b b r m m d r      y  c    n  a d v v d           d
" c m b a h s o r a e p p c a r     c  y    n  r r s s a d d d   e c
" l c c c c e l e l a c c e c e     l  s n  e  e e b b l c r w e n l
" k s s s s l h q e d l 0 n k s    e  t p  r  g g a d e s d r n a k
 [c,x,x,x,x,x,x,x,x,x,0,0,0,x,0]->[ls,x,0,t4r,0,0,1,0,0,0,1,0,1,1,0];"18
 [c,x,x,x,x,x,x,x,x,x,1,1,0,x,0]->[ls,x,0,t0 ,0,0,1,0,0,0,0,1,1,0,0];
 [c,x,x,x,x,x,x,x,0,x,x,x,0,x,0]->[ls,x,0,t0 ,0,0,1,0,0,0,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,1,1,x,x,0,x,0]->[ls,x,0,t1r,0,0,1,0,1,0,1,0,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,0,x,0]->[ls,x,0,t2r,0,0,1,0,0,1,1,0,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,0,x,0]->[ls,x,0,t3r,0,0,1,0,0,1,1,0,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,0,x,0]->[ls,x,0,t3r,0,0,1,0,0,1,1,0,1,1,0];
 [0,x,x,x,x,x,x,x,x,x,x,x,1,x,0]->[ls,x,0,t3r,0,0,1,0,0,1,1,0,1,1,1];
 [c,x,x,x,x,x,x,x,x,x,x,x,1,x,0]->[ls,x,0,t4r,0,0,1,0,0,1,1,0,1,1,1];
 [c,x,x,x,x,x,x,x,x,x,0,0,0,x,0]->[ls,x,0,t4r,0,0,1,0,0,0,1,0,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,1,0,0,x,0]->[ls,x,1,t7w,0,0,1,0,0,0,0,1,1,0,0];
 [c,0,0,0,0,0,0,0,x,x,x,x,0,x,0]->[wa,1,0,t0 ,0,0,0,0,0,0,0,1,0,0,0];
 [c,0,0,0,0,0,0,1,x,x,x,x,0,x,0]->[ls,1,0,t0 ,0,0,1,0,0,0,0,1,1,0,0];
 [c,x,x,x,x,x,x,x,0,x,x,x,0,x,0]->[ls,1,0,t0 ,0,0,1,0,0,0,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,1,0,x,x,0,x,0]->[ls,x,0,t1w,0,0,1,1,1,0,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,0,x,0]->[ls,x,0,t2w,0,0,1,1,0,1,0,1,1,1,0];"33
"           d       v
"   d       r   v v s     d                   o o o o
"   r v   e a   s s b j j r d      c       e  e e e e
"   a s c t m h b b r m m d r      y  c    n  a d v v d           d
" c m b a h s o r a e p p c a r    c  y    n  r r s s a d d d   e c
" l c c c c e l e l a c c e c e    l  s n  e  e e b b l c r w e n l
" k s s s s l h q e d l 0 n k s    e  t p  r  g g a d e s d r n a k
 [c,x,x,x,x,x,x,x,x,x,x,x,0,x,0]->[ls,x,0,t3w,0,0,1,1,0,1,0,1,1,1,0];"34
 [c,x,x,x,x,x,x,x,x,x,x,x,0,0,0]->[ls,x,0,t3w,0,0,1,1,0,1,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,0,1,0]->[ls,x,0,t4w,0,0,1,1,0,1,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,0,0,0,x,0]->[ls,x,0,t4w,0,0,1,1,0,0,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,1,0,x,0]->[ls,x,0,t1w,0,0,1,1,1,0,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,0,x,0]->[ls,x,0,t2w,0,0,1,1,0,1,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,0,x,0]->[ls,x,0,t3w,0,0,1,1,0,1,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,0,0,0]->[ls,x,0,t3w,0,0,1,1,0,1,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,0,1,0]->[ls,x,0,t4w,0,0,1,1,0,1,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,0,0,0,x,0]->[ls,x,0,t4w,0,0,1,1,0,0,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,1,1,0,x,0]->[ls,x,0,t0 ,0,0,1,0,0,0,0,1,1,0,0];
 [c,x,x,x,x,x,x,x,1,0,x,x,0,x,0]->[ls,x,0,t1w,0,0,1,1,1,0,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,0,x,0]->[ls,x,0,t2w,0,0,1,1,0,1,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,0,x,0]->[ls,x,0,t3w,0,0,1,1,0,1,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,x,x,0,0,0]->[ls,x,0,t3w,0,0,1,1,0,1,0,1,1,1,0];"48
"           d       v
"   d       r   v v s     d                   o o o o
"   r v   e a   s s b j j r d      c       e  e e e e
"   a s c t m h b b r m m d r      y  c    n  a d v v d           d
" c m b a h s o r a e p p c a r    c  y    n  r r s s a d d d   e c
" l c c c c e l e l a c c e c e    l  s n  e  e e b b l c r w e n l
" k s s s s l h q e d l 0 n k s    e  t p  r  g g a d e s d r n a k
 [c,x,x,x,x,x,x,x,x,x,x,x,0,1,0]->[ls,x,0,t4w,0,0,1,1,0,1,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,0,0,0,x,0]->[ls,x,0,t4w,0,0,1,1,0,0,0,1,1,1,0];
 [c,x,x,x,x,x,x,x,x,x,1,0,0,x,0]->[ls,x,1,t7w,0,0,1,0,0,0,0,1,1,0,0];
 [c,0,0,0,0,0,0,0,x,x,x,x,0,x,0]->[wa,1,0,t0 ,0,0,0,0,0,0,0,1,0,0,0];

"######################################################################
"           test vectors for cpu access to vsb.
"######################################################################
```

```
test_vectors
([clk,dramcs,vsbcs,cacs,ethcs,dramcs ,
hold,vsbreq,mrq,subwr,jmpc1,jmpc0,drdcen,drack,res]
 -> [cycle,cyst,np,enner,oeareg,oedreg,oevsba,oevsbd,crdcen,cack,
     vsbaclk,mast,start,dle ,dle]);

"           d
"   d       r   v v       d                   o o o o c   v
"   r v   e a   s s s j j r d      c       e  e e e e r   b   s
"   a s c t m h b b u m m d r      y  c    n  a d v v d c a m t r d
" c m b a h s o r a b p p c a r    c  y    n  r r s s c a c a a e c
" l c c c c e l e l w c c e c e    l  s n  e  e e b b e c l s r a l
" k s s s s l h q e r 1 0 n k s    e  t p  r  g g a d n k k t t d k
 [c,x,x,x,x,x,x,x,x,x,x,x,0,0,1]->[wa,1,0,t0 ,0,0,0,0,0,0,0,0,0,0,0];"53
 [c,0,1,0,0,0,x,x,x,0,x,x,0,0,0]->[lv,1,0,t0 ,1,0,0,0,0,0,0,1,0,0,0];
 [c,0,1,0,0,0,x,x,x,0,x,x,0,0,0]->[lv,0,0,t1r,1,0,0,1,0,0,1,1,0,1,0];
 [c,0,1,0,0,0,x,x,x,x,x,x,0,0,0]->[lv,0,0,t2r,1,0,0,1,0,0,0,1,1,1,0];
 [c,0,1,0,0,0,x,x,x,x,x,0,0,0,0]->[lv,0,0,t2r,1,0,0,1,0,0,0,1,1,1,0];
 [c,0,1,0,0,0,x,x,x,x,x,1,0,0,0]->[lv,0,1,t3r,1,0,0,1,0,0,0,1,1,1,1];
 [c,0,1,0,0,0,0,0,x,x,x,x,0,0,0]->[wa,0,0,t0 ,0,0,0,1,1,0,0,0,0,0,0];
 [c,0,0,0,0,0,0,0,x,x,x,x,0,0,0]->[wa,1,0,t0 ,0,0,0,0,0,0,0,0,0,0,0];"60


"           d
"   d       r   v v       d                   o o o o     v
"   r v   e a   s s s j j r d      c       e  e e e e     b   s
"   a s c t m h b b u m m d r      y  c    n  a d v v d   a m t r d
" c m b a h s o r a b p p c a r    c  y    n  r r s s a d c a a e c
" l c c c c e l e l w c c e c e    l  s n  e  e e b b l c l s r a l
" k s s s s l h q e r 1 0 n k s    e  t p  r  g g a d e s k t t d k
 [c,0,0,0,0,0,0,0,x,x,x,x,0,0,0]->[wa,1,0,t0 ,0,0,0,0,0,0,0,0,0,0,0];"61
 [c,0,1,0,0,0,0,0,x,1,x,x,0,0,0]->[lv,1,0,t0 ,1,0,0,0,0,0,0,1,0,0,0];
 [c,0,1,0,0,0,0,0,x,x,x,x,0,0,0]->[lv,0,0,t1w,1,1,0,0,0,0,1,1,0,0,1];
 [c,0,1,0,0,0,0,0,x,x,x,x,0,0,0]->[lv,0,0,t2w,1,1,0,0,0,0,0,1,1,0,0];
 [c,0,1,0,0,0,0,0,x,x,x,0,0,0,0]->[lv,0,0,t2w,1,1,0,0,0,0,0,1,1,0,0];
 [c,0,1,0,0,0,0,0,x,x,x,1,0,0,0]->[lv,0,1,t3w,1,1,0,0,0,0,0,1,1,0,0];
 [c,0,0,0,0,0,0,0,x,x,x,x,0,0,0]->[wa,1,0,t0 ,0,0,0,0,0,0,0,0,0,0,0];"67

"#######################################################################
"        test vectors for ethernet access to memory.
"#######################################################################

test_vectors
([clk,dramcs,vsbcs,cacs,ethcs,dramcs ,
hold,vsbreq,lock,ads,reade,blast,drdcen,drack,res]
 -> [cycle,cyst,np,enner,oeareg,oedreg,oevsba,oevsbd,
     holda,erdy,ebrdy,dale,dcs,dwr,dwr,burstdm,dle]);

"          d                                                       b
"  d       r   v         d                   o o o o               u
"  r v   e a   s     r b r d      c       e  e e e e h   e         r
"  a s c t m h b l   e l d r      y  c    n  a d v v o e b d       s d
"c m b a h s o r o a a a c a r    c  y    n  r r s s l r r a d d d t c
"l c c c c e l e c d d s e c e    l  s n  e  e e b b d d d l c r w d l
"k s s s s l h q k s e t n k s    e  t p  r  g g a d a y y e s d d m k
[c,x,x,x,x,x,x,x,x,x,x,x,0,x,1]->[wa,1,0,t0 ,0,0,0,0,0,0,0,0,0,0,1,0,0];"68
```

```
[c,0,0,0,0,0,0,0,x,x,x,x,0,x,0]->[wa,1,0,t0 ,0,0,0,0,0,0,0,0,0,0,1,0,0];
[c,0,0,0,0,0,1,x,x,x,x,x,0,x,0]->[le,1,0,t0 ,0,0,0,0,1,0,0,0,0,0,1,1,0];
[c,0,0,0,0,0,1,x,x,0,x,x,0,x,0]->[le,1,0,t0 ,0,0,0,0,1,0,0,0,0,0,1,1,0];
[c,0,0,0,0,0,1,x,x,1,1,x,0,x,0]->[le,1,0,t1r,0,0,0,0,1,0,0,1,0,1,0,1,0];
[c,0,0,0,0,0,1,x,x,x,x,x,0,x,0]->[le,1,0,t2r,0,0,0,0,1,0,0,0,1,1,0,1,0];
[c,0,0,0,0,0,1,x,x,x,x,0,0,x,0]->[le,1,0,t3r,0,0,0,0,1,0,1,0,1,1,0,1,0];
[c,0,0,0,0,0,1,x,x,x,x,x,0,x,0]->[le,1,0,t3r,0,0,0,0,1,0,1,0,1,1,0,1,0];
[c,0,0,0,0,0,1,x,x,x,x,x,1,x,0]->[le,1,0,t4r,0,0,0,0,1,0,1,0,1,1,0,1,1];
[c,0,0,0,0,0,1,x,x,x,x,x,0,x,0]->[le,1,0,t4r,0,0,0,0,1,0,1,0,1,1,0,1,0];
[c,0,0,0,0,0,1,x,x,x,x,x,1,x,0]->[le,1,0,t5r,0,0,0,0,1,0,1,0,1,1,0,1,1];
[c,0,0,0,0,0,1,x,x,x,x,x,0,x,0]->[le,1,0,t5r,0,0,0,0,1,0,1,0,1,1,0,1,0];
[c,0,0,0,0,0,1,x,x,x,x,x,1,x,0]->[le,1,0,t6r,0,0,0,0,1,1,0,0,1,1,0,1,1];
[c,0,0,0,0,0,1,x,x,x,x,x,0,x,0]->[le,1,0,t6r,0,0,0,0,1,1,0,0,1,1,0,1,0];
[c,0,0,0,0,0,1,x,x,x,x,x,1,x,0]->[le,1,1,t0 ,0,0,0,0,1,1,0,0,0,0,1,1,1];
[c,0,0,0,0,0,1,x,1,0,x,x,0,x,0]->[le,1,0,t0 ,0,0,0,0,1,0,0,0,0,0,1,1,0];"83
"          d                                                       b
"  d       r   v         d                   o o o o               u
"  r v   e a   s     r b r d       c       e  e e e e h   e         r
"  a s c t m h b l   e l d r       y  c    n  a d v v o e b d       s d
"c m b a h s o r o a a a c a r    c  y    n  r r s s l r r a d d d t c
"l c c c c e l e c d d s e c e    l  s n  e  e e b b d d d l c r w d l
"k s s s s l h q k s e t n k s    e  t p  r  g g a d a y y e s d d m k
[c,0,0,0,0,0,1,x,x,0,x,x,0,x,0]->[le,1,0,t0 ,0,0,0,0,1,0,0,0,0,0,1,1,0];"84
[c,0,0,0,0,0,1,x,x,1,1,x,0,x,0]->[le,1,0,t1r,0,0,0,0,1,0,0,1,0,1,0,1,0];
[c,0,0,0,0,0,1,x,x,x,x,x,0,x,0]->[le,1,0,t2r,0,0,0,0,1,0,0,0,1,1,0,1,0];
[c,0,0,0,0,0,1,x,x,x,x,1,0,x,0]->[le,1,0,t6r,0,0,0,0,1,1,0,0,1,1,0,1,0];
[c,0,0,0,0,0,1,x,x,x,x,x,0,x,0]->[le,1,0,t6r,0,0,0,0,1,1,0,0,1,1,0,1,0];
[c,0,0,0,0,0,1,x,x,x,x,x,1,x,0]->[le,1,1,t0 ,0,0,0,0,1,1,0,0,0,0,1,1,1];
[c,0,0,0,0,0,1,x,x,0,x,x,0,x,0]->[le,1,0,t0 ,0,0,0,0,1,0,0,0,0,0,1,1,0];
[c,0,0,0,0,0,1,x,x,0,x,x,0,x,0]->[le,1,0,t0 ,0,0,0,0,1,0,0,0,0,0,1,1,0];
[c,0,0,0,0,0,1,x,x,1,0,x,0,x,0]->[le,1,0,t1w,0,0,0,0,1,0,0,1,0,0,1,1,0];
[c,0,0,0,0,0,1,x,x,x,x,x,0,x,0]->[le,1,0,t2w,0,0,0,0,1,0,0,0,1,0,1,1,0];
[c,0,0,0,0,0,1,x,x,x,x,0,0,x,0]->[le,1,0,t3w,0,0,0,0,1,0,1,0,1,0,1,1,0];
[c,0,0,0,0,0,1,x,x,x,x,x,0,0,0]->[le,1,0,t3w,0,0,0,0,1,0,1,0,1,0,1,1,0];
[c,0,0,0,0,0,1,x,x,x,x,x,0,1,0]->[le,1,0,t4w,0,0,0,0,1,0,1,0,1,0,1,1,0];
[c,0,0,0,0,0,1,x,x,x,x,x,0,0,0]->[le,1,0,t4w,0,0,0,0,1,0,1,0,1,0,1,1,0];
[c,0,0,0,0,0,1,x,x,x,x,x,0,1,0]->[le,1,0,t5w,0,0,0,0,1,0,1,0,1,0,1,1,0];"98
"          d                                                       b
"  d       r   v         d                   o o o o               u
"  r v   e a   s     r b r d       c       e  e e e e h   e         r
"  a s c t m h b l   e l d r       y  c    n  a d v v o e b d       s d
"c m b a h s o r o a a a c a r    c  y    n  r r s s l r r a d d d t c
"l c c c c e l e c d d s e c e    l  s n  e  e e b b d d d l c r w d l
"k s s s s l h q k s e t n k s    e  t p  r  g g a d a y y e s d d m k
[c,x,x,x,x,x,1,x,x,x,x,x,0,0,0]->[le,1,0,t5w,0,0,0,0,1,0,1,0,1,0,1,1,0];"99
[c,x,x,x,x,x,1,x,x,x,x,x,0,1,0]->[le,x,0,t6w,0,0,0,0,1,1,0,0,1,0,1,1,0];
[c,x,x,x,x,x,1,x,x,x,x,x,0,0,0]->[le,x,0,t6w,0,0,0,0,1,1,0,0,1,0,1,1,0];
[c,x,x,x,x,x,1,x,x,x,x,x,0,1,0]->[le,x,1,t0 ,0,0,0,0,1,1,0,0,0,0,1,1,0];
[c,0,0,0,0,0,0,0,0,x,x,x,0,x,0]->[wa,1,0,t0 ,0,0,0,0,0,0,0,0,0,0,1,0,0];
[c,0,0,0,0,0,1,0,x,x,x,x,0,x,0]->[le,1,0,t0 ,0,0,0,0,1,0,0,0,0,0,1,1,0];
[c,x,x,x,x,x,1,x,x,1,0,x,0,x,0]->[le,x,0,t1w,0,0,0,0,1,0,0,1,0,0,1,1,0];
[c,x,x,x,x,x,1,x,x,x,x,x,0,x,0]->[le,x,0,t2w,0,0,0,0,1,0,0,0,1,0,1,1,0];
[c,x,x,x,x,x,1,x,x,x,x,1,0,x,0]->[le,x,0,t6w,0,0,0,0,1,1,0,0,1,0,1,1,0];
[c,x,x,x,x,x,1,x,x,x,x,x,0,0,0]->[le,x,0,t6w,0,0,0,0,1,1,0,0,1,0,1,1,0];
[c,x,x,x,x,x,1,x,x,x,x,x,0,1,0]->[le,x,1,t0 ,0,0,0,0,1,1,0,0,0,0,1,1,0];
[c,0,0,0,0,0,0,0,0,x,x,x,0,x,0]->[wa,1,0,t0 ,0,0,0,0,0,0,0,0,0,0,1,0,0];"11
0
```

```
"#########################################################################
"           test vectors for cpu access to memory
"#########################################################################


test_vectors
([clk,dramcs,vsbcs,cacs,ethcs,dramcs ,
hold,vsbreq,subwr,burstc,drdcen,drack,res]
 -> [cycle,cyst,np,enner,oeareg,oedreg,oevsba,oevsbd,
     dale,dcs,dwr,dwr,dburst,dle,crdcen,cack]);

"test vectors for cpu access to memory
"          d
"  d       r   v   b d                   o o o o         d   c
"  r v   e a   s s u r d       c       e  e e e e         b   r
"  a s c t m h b u r d r       y  c    n  a d v v d       u d d c
"c m b a h s o r b s c a r     c  y    n  r r s s a d d d r c c a
"l c c c c e l e w t e c e     l  s n  e  e e b b l c r w s l e c
"k s s s s l h q r c n k s     e  t p  r  g g a d e s d d t k n k
[c,x,x,x,x,x,x,x,x,x,0,x,1]->[wa,1,0,t0 ,0,0,0,0,0,0,0,1,0,0,0,0];"111
[c,1,0,0,0,0,x,x,x,x,0,x,0]->[lm,1,0,t0 ,1,0,0,0,0,0,0,1,0,0,0,0];
[c,1,0,0,0,0,x,x,0,x,0,x,0]->[lm,0,0,t1r,1,0,0,0,1,0,1,0,0,0,0,0];
[c,1,0,0,0,0,x,x,x,1,0,x,0]->[lm,0,0,t2r,1,0,0,0,0,1,1,0,1,0,0,0];
[c,1,0,0,0,0,x,x,x,1,0,x,0]->[lm,0,0,t3r,1,0,0,0,0,1,1,0,1,0,0,0];
[c,1,0,0,0,0,x,x,x,1,0,x,0]->[lm,0,0,t3r,1,0,0,0,0,1,1,0,1,0,0,0];
[c,1,0,0,0,0,x,x,x,1,1,x,0]->[lm,0,0,t4r,1,0,0,0,0,1,1,0,1,1,1,0];
[c,1,0,0,0,0,x,x,x,1,0,x,0]->[lm,0,0,t4r,1,0,0,0,0,1,1,0,1,0,0,0];
[c,1,0,0,0,0,x,x,x,1,1,x,0]->[lm,0,0,t5r,1,0,0,0,0,1,1,0,1,1,1,0];
[c,1,0,0,0,0,x,x,x,1,0,x,0]->[lm,0,0,t5r,1,0,0,0,0,1,1,0,1,0,0,0];
[c,1,0,0,0,0,x,x,x,1,1,x,0]->[lm,0,0,t6r,1,0,0,0,0,1,1,0,1,1,1,0];
[c,1,0,0,0,0,x,x,x,1,0,x,0]->[lm,0,0,t6r,1,0,0,0,0,1,1,0,1,0,0,0];
[c,1,0,0,0,0,0,0,x,1,1,x,0]->[lm,0,1,t7w,1,0,0,0,0,1,0,1,1,1,1,0];
[c,0,0,0,0,0,0,0,x,x,x,x,0]->[wa,1,0,t0 ,0,0,0,0,0,0,0,1,0,0,0,0];"124
"          d
"  d       r   v   b d                   o o o o         d   c
"  r v   e a   s s u r d       c       e  e e e e         b   r
"  a s c t m h b u r d r       y  c    n  a d v v d       u d d c
"c m b a h s o r b s c a r     c  y    n  r r s s a d d d r c c a
"l c c c c e l e w t e c e     l  s n  e  e e b b l c r w s l e c
"k s s s s l h q r c n k s     e  t p  r  g g a d e s d d t k n k
[c,1,0,0,0,0,x,x,x,x,0,x,0]->[lm,1,0,t0 ,1,0,0,0,0,0,0,1,0,0,0,0];"125
[c,1,0,0,0,0,x,x,0,0,0,x,0]->[lm,0,0,t1r,1,0,0,0,1,0,1,0,0,0,0,0];
[c,1,0,0,0,0,x,x,x,0,0,x,0]->[lm,0,0,t2r,1,0,0,0,0,1,1,0,0,0,0,0];
[c,1,0,0,0,0,x,x,x,0,0,x,0]->[lm,0,0,t6r,1,0,0,0,0,1,1,0,0,0,0,0];
[c,1,0,0,0,0,x,x,x,x,0,x,0]->[lm,0,0,t6r,1,0,0,0,0,1,1,0,0,0,0,0];
[c,1,0,0,0,0,0,0,x,x,1,x,0]->[lm,0,1,t7w,1,0,0,0,0,1,0,1,0,1,1,0];
[c,0,0,0,0,0,0,0,x,x,0,x,0]->[wa,1,0,t0 ,0,0,0,0,0,0,0,1,0,0,0,0];
[c,1,0,0,0,0,0,0,x,x,0,x,0]->[lm,1,0,t0 ,1,0,0,0,0,0,0,1,0,0,0,0];
[c,1,0,0,0,0,x,x,1,x,0,x,0]->[lm,0,0,t1w,1,0,0,0,1,0,0,1,0,0,0,0];
[c,1,0,0,0,0,x,x,x,x,0,x,0]->[lm,0,0,t2w,1,1,0,0,0,1,0,1,0,0,0,1];
[c,1,0,0,0,0,x,x,x,x,0,x,0]->[lm,0,0,t3w,1,1,0,0,0,1,0,1,0,0,0,0];
[c,1,0,0,0,0,x,x,x,x,0,0,0]->[lm,0,0,t3w,1,1,0,0,0,1,0,1,0,0,0,0];
[c,1,0,0,0,0,x,x,x,x,0,1,0]->[lm,0,1,t7w,1,1,0,0,0,1,0,1,0,0,0,0];
[c,0,0,0,0,0,0,0,x,x,0,x,0]->[wa,1,0,t0 ,0,0,0,0,0,0,0,1,0,0,0,0];
```

```
end cc0
```

Supermax Field Change Notice no. 158

Module:	BAIO301, BAIO302 Basic I/O module.
Date:	96-01-25

Category:

- Production change.
- In the field: When a TPE301 submodule is installed.

Corrects the error:

The submodule interface locks up, when TPE301 acts as a master on the interface. HDLC301 and SCU302 are slave devices only. The operation of these submodules is independent of this error.
Symptoms: The submodule interface stops. System crash caused by time out on BAIO30x.

Needed tools:

68 pin PLCC extraction tool. 44 pin PLCC extraction tool. 20 pin PLCC extraction tool.
Supermax FCN kit 158, stock number 95101580, consisting of:

1. One MACH labeled "**BIO261**".
2. One MACH labeled "**BIO272**".
3. One PROM labeled "**FCN158**".

Description:

- All previous FCNs for this module must be made.
- Replace MACH in position U180 with MACH labeled "**BIO261**". This circuit is labeled "**BIO260**" on BAIO301 and BAIO302.
- Replace MACH in position U186 with MACH labeled "**BIO272**". This circuit is labeled "**BIO271**" on BAIO301 and BAIO302.
- Replace FCN PROM in position U204 with new FCN PROM labeled "**FCN158**". This circuit is labeled "**FCN156**" on BAIO301 and BAIO302.
- Do not update the revision field. On BAIO301 and BAIO302 the field should be **REV: ABCDEFGHIJK**

Circuits involved:

Name	Type
U180	MACH210-15JC
U186	MACH220-15JC
U204	AM27519AJC

Previous FCN:

BAIO301: FCN156.
BAIO302: FCN156.

Frigivelse
af
Programmerbare Komponenter

Modulnavn: BAIO301, BAIO302

Dato: 960104

Initialer: kan

Note: FCN158

Filnavn	Checksum	Label	Type	Råvarenummer
bio261.jed	00037F85	BIO261	MACH210-15	99.010.905
bio272.jed	000605E7	BIO272	MACH220-15	99.010.908

PAL list

Part no:	750000	Module:	BAIO302-0	Pcb no:	
Status:	Final	Date:	960125	Page:	1 / 1
Init:	KAN	FCN158			

Label	Position	Part no.	Type	File	Checksum
BIO000	U193	99010021	PAL16R4D/2JC	bio000.jed	6D57
BIO010	U64	99010023	PAL16R6D/2JC	bio010.jed	1B8E
BIO020	U92	99010025	PAL16L8-5JC	bio020.jed	358C
BIO030	U54	99010020	PAL16R4-7JC	bio030.jed	1E15
BIO041	U175	99010025	PAL16L8-5JC	bio041.jed	44FA
BIO050	U189	99010907	MACH220-12JC	bio050.jed	57CF
BIO060	U205	99010900	MACH110-12JC	bio060.jed	1237
BIO070	U23	99010906	MACH210-20JC	bio070.jed	6878
BIO080	U26	99010027	PAL16L8D-2JC	bio080.jed	5DC4
BIO090	U28	99010021	PAL16R4D/2JC	bio090.jed	1EB8
BIO100	U40	99010021	PAL16R4D/2JC	bio100.jed	1EA3
BIO112	U57	99010908	MACH220-15JC	bio112.jed	CCBA
BIO121	U59	99010026	PAL16L8-7JC	bio121.jed	1DBE
BIO130	U58	99010020	PAL16R4-7JC	bio130.jed	2DC7
BIO140	U72	99010908	MACH220-15JC	bio140.jed	995B
BIO150	U73	99010026	PAL16L8-7JC	bio150.jed	4A1A
BIO160	U75	99010027	PAL16L8D-2JC	bio160.jed	7C30
BIO171	U78	99010908	MACH220-15JC	bio171.jed	15EB
BIO180	U79	99010027	PAL16L8D-2JC	bio180.jed	2A9F
BIO193	U108	99010908	MACH220-15JC	bio193.jed	903A
BIO201	U110	99010904	MACH210-12JC	bio201.jed	69BC
BIO200	U147	99010904	MACH210-12JC	bio200.jed	F0FF
BIO210	U113	99010024	PAL16R8D/2JC	bio210.jed	4DEB
BIO221	U115	99010904	MACH210-12JC	bio221.jed	35BA
BIO230	U138	99010908	MACH220-15JC	bio230.jed	2D7F
BIO250	U148	99010904	MACH210-12JC	bio250.jed	457D
BIO261	U180	99010905	MACH210-15JC	bio261.jed	7F58
BIO272	U186	99010908	MACH220-15JC	bio272.jed	05E7
BIO281	U187	99010026	PAL16L8-7JC	bio281.jed	17B2
BIO290	U188	99010908	MACH220-15JC	bio290.jed	E55D
BIO300	U229	99010216	PAL20R8-5JC	bio300.jed	DEF0
BIO310	U232	99010022	PAL16R6-5JC	bio310.jed	2BE2
BIO320	U194	99010902	MACH110-20JC	bio320.jed	B930
BIO330	U228	99010216	PAL20R8-5JC	bio330.jed	7377
BIO341	U195	99010904	MACH210-12JC	bio341.jed	E6A1
BIO350	U202	99010903	MACH120-15JC	bio350.jed	73FC
BIO360	U236	99010902	MACH110-20JC	bio360.jed	DD36
BIO370	U241	99010902	MACH110-20JC	bio370.jed	604D
BIO380	U245	99010604	PALCE16V8H-7JC/5	bio380.jed	234F *
BIO391	U35	99010021	PALCE16V8H-7JC/5	bio391.jed	3718
IDxxxx	U203	99012095	82S123AA		
FCN	U204	99012095	82S123AA		

* as PAL16R6

All JEDEC files are located on nessie at : /wd/mudv3/pal/baio302-0/jed
All ABL files are located on nessie at : /wd/mudv3/pal/baio302-0/abl
Files for generating ID and FCN PROMs are generated in the fab.

Supermax Field Change Notice no. 156

Module:	BAIO301, BAIO302 Basic I/O module.
Date:	95-10-12. DRAFT

Category:

- Production change.
- In the field: Without any unnecessary delay.

Corrects the error:

Error in data transfer between SCSI data buffer and SCSI agent. Under certain circumstances the control logic starts simultaneous read and write block transfers.
Symptoms: SCSI channel stops. System crash caused by time out on BAIO30x or parity error on BAIO30x. Write data from SCSI is written in a wrong address in Global Memory.

Needed tools:

68 pin PLCC extraction tool.
Supermax FCN kit 156, stock number 95101560, consisting of:

1. One MACH labeled "**BIO193**".
2. One PROM labeled "**FCN156**".

Description:

- All previous FCNs for this module must be made. Replace MACH in position U108 with MACH labeled "**BIO193**". This circuit is labled "**BIO191**" on BAIO301 and "**BIO192**" on BAIO302.
- Replace FCN PROM in position U204 with new FCN PROM labeled "**FCN156**". This circuit is labled "**FCN146**" on BAIO301 and "**FCN155**" on BAIO302.
- Do not update the revision field. On BAIO301 and BAIO302 the field should be **REV: ABCDEFGHIJK**

Circuits involved:

Name	Type
U108	MACH220-15JC
U204	83S123AA

Previous FCN:

BAIO301: FCN146.
BAIO302: FCN155.

Frigivelse
af
Programmerbare Komponenter

Modulnavn:	BAI0302-0
Dato:	951012
Initialer:	kan
Note:	FCN156

Filnavn	Checksum	Label	Type	Råvarenummer
bio193.jed	0005903A	BIO193	MACH220-15	99010908

Supermax Field Change Notice no. 155

Module:	BAIO302
Date:	95-10-02. DRAFT

Category:

- Production change.
- In the field: when error occurs.

Corrects the error:

Clock skew between processor and memory controller. The memory controller will miss memory requests from the processor.
BAIO cannot boot. BAIO cannot execute memory test program. BAIO stops.

Needed tools:

44 pin PLCC extraction tool.
Supermax FCN kit 155, stock number 95101550, consisting of:

1. One MACH labeled "**BIO201**".
2. One MACH labeled "**BIO221**".
3. One PROM labeled "**FCN155**".

Description:

- Replace MACH labeled "**BIO200**" in position U110 with MACH labeled "**BIO201**".
- Replace MACH labeled "**BIO220**" in position U115 with MACH labeled "**BIO221**".
- Replace FCN PROM in position U204 with new FCN PROM labeled "**FCN155**".
- Do not update the revision field. Field should be **REV: ABCDEFGHIJK**

Circuits involved:

Name	Type
U110	MACH210-12
U115	MACH210-12
U204	83S123AA

Previous FCN:

None

```
mailx -s"FCN155 til BAIO302" kkj <sendl

Til:       Produktionen
Fra:       Knud Arne Nielsen
Dato:      951002
Ang.:      FCN155 til BAIO302
------------------------------------------------------------------
Sendt elektronisk via tftp



tftp -i paxmax put ./bio201.abl /tmp/bio201.abl
tftp -i paxmax put ./bio221.abl /tmp/bio221.abl
tftp -i paxmax put ./bio201.jed /tmp/bio201.jed
tftp -i paxmax put ./bio221.jed /tmp/bio221.jed

tftp -i paxmax put ./fril /tmp/fril Frigivelse

FCN155 følger på papir.
```

Frigivelse
af
Programmerbare Komponenter

Modulnavn:	BAIO302
Dato:	951002
Initialer:	kan

Filnavn	Checksum	Label	Type	Råvarenummer
bio201.jed	000469BC	BIO201	MACH210-12	
bio221.jed	000435BA	BIO221	MACH210-12	

dde Dansk Data Elektronik A/S

Initials	Page	
Date	Project	

	skal være	er	
1	GND	GND	
2	C+	NC	
3	TX+	C÷	
4	GND	C+	
5	RX+	TRM −	(= TX−)
6	GND	TRM +	(= TX+)
7	~~NC~~ TERM	NC	
8	GND	NC	
9	C−	RCV−	
10	TX−	RCV+	
11	GND	12V	
12	RX−	GND	
13	12V	TERM	
14	GND	TERM	
15	~~NC~~ TERM	NC	

Steve,

This is the Interleaved Design for a DRAM subsystem around the R3051 at 33Mhz.

The Design is not complete yet & there might be errors in the timings or the logic, but the overall idea is correct

Please pass on this information to your Customer & let me know if you need any more help.

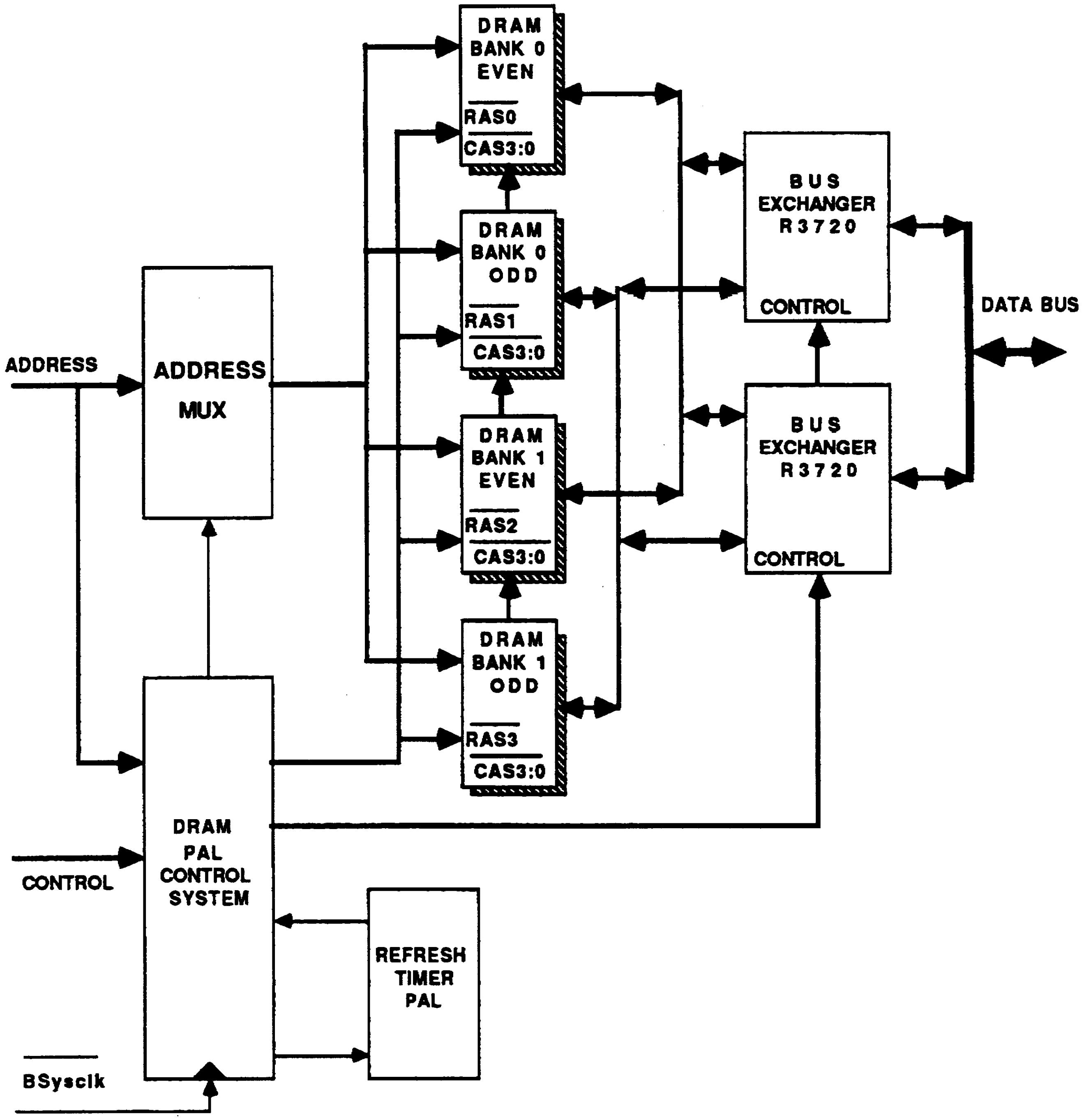
Bob Napaa.

Note:-

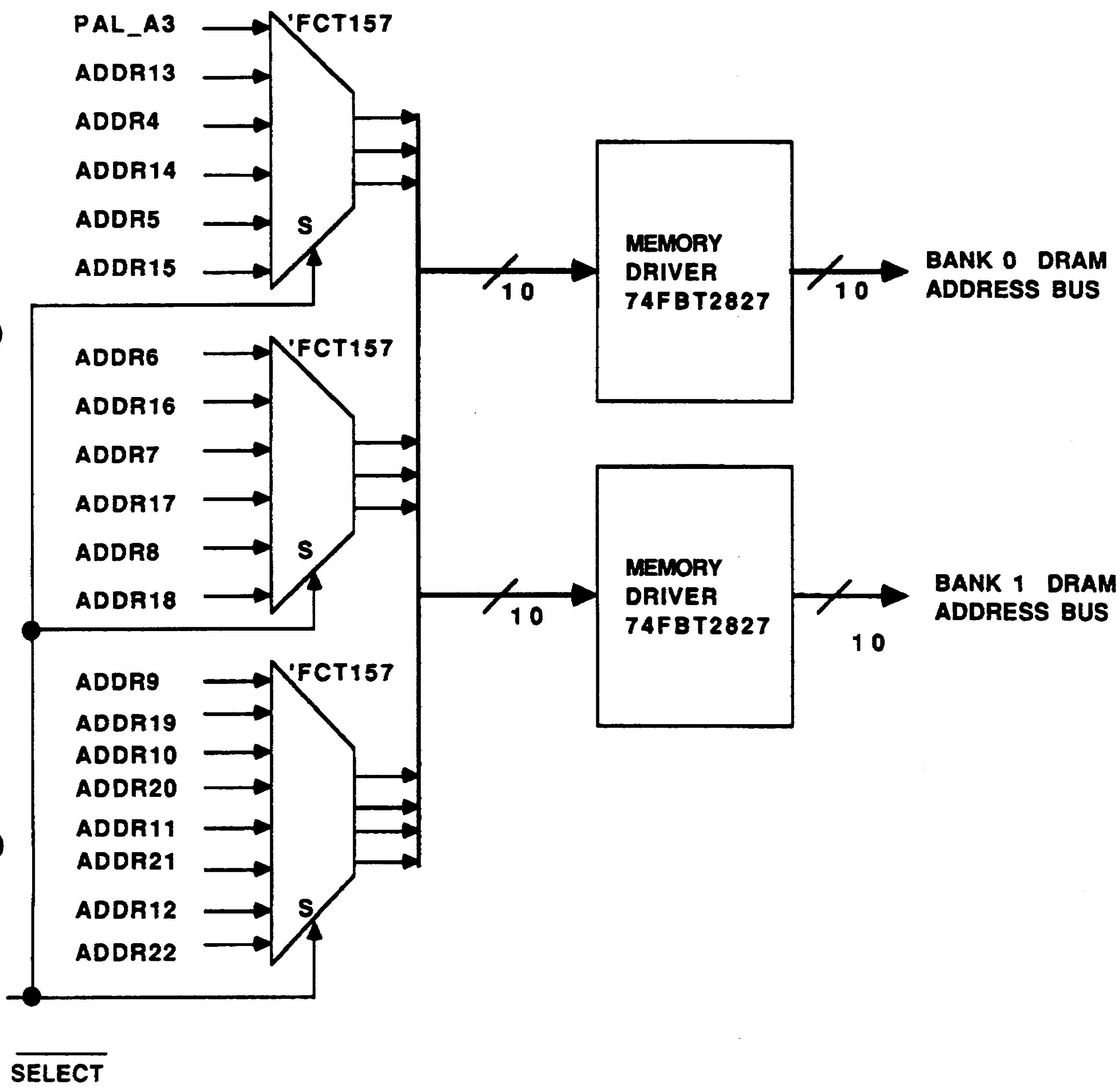
① The Customer will need to adjust the timings & the logic for the 25 mhz design.

② this Design is very similar to the App Note I wrote about non-Interleaved DRAM subsystems.

③ The DRAM Controller manual (Chp 9 & 10) describes interleaved DRAM systems.



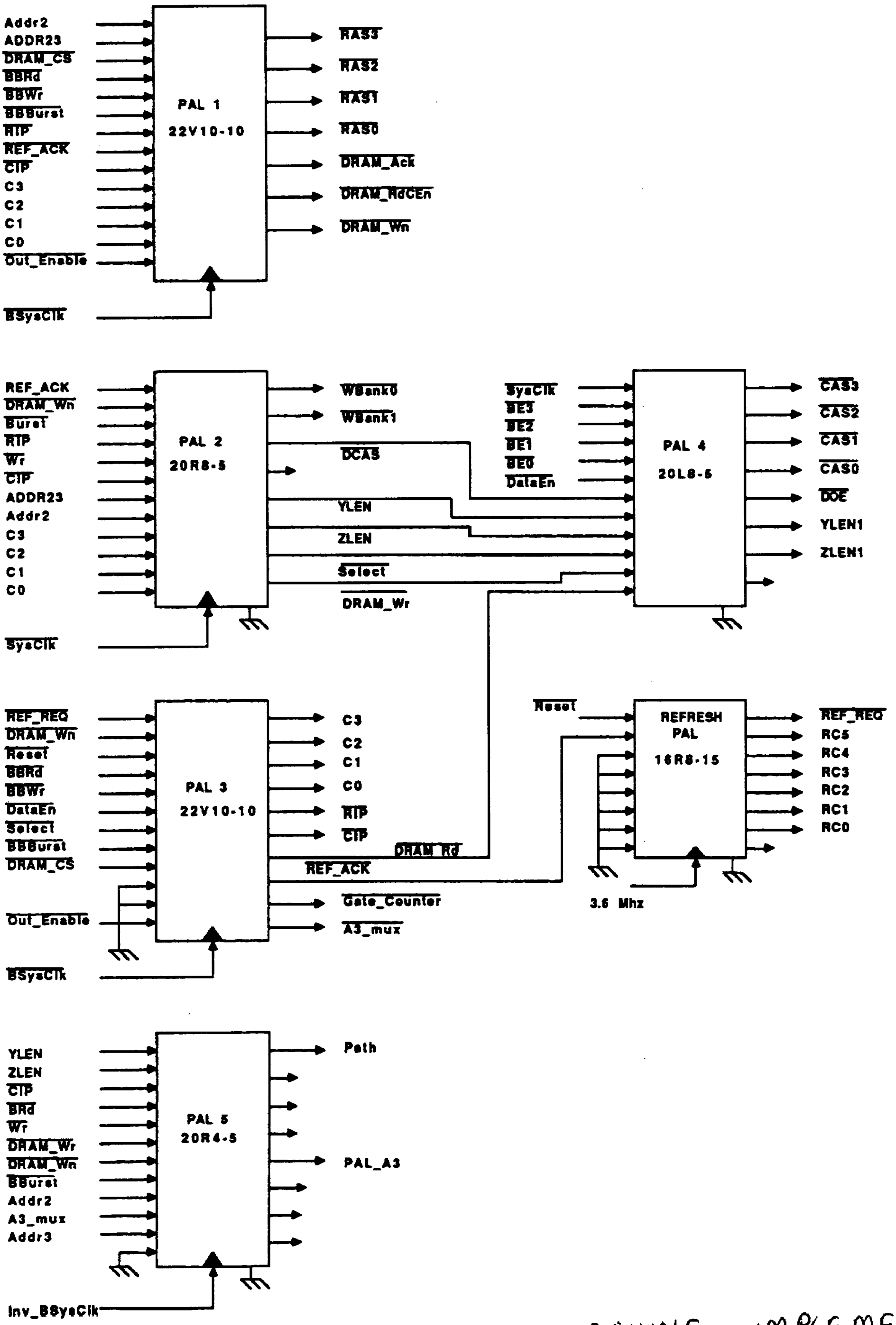
OVERALL BLOCK DIAGRAM OF ANY INTERLEAVED DRAM MEMORY SYSTEM



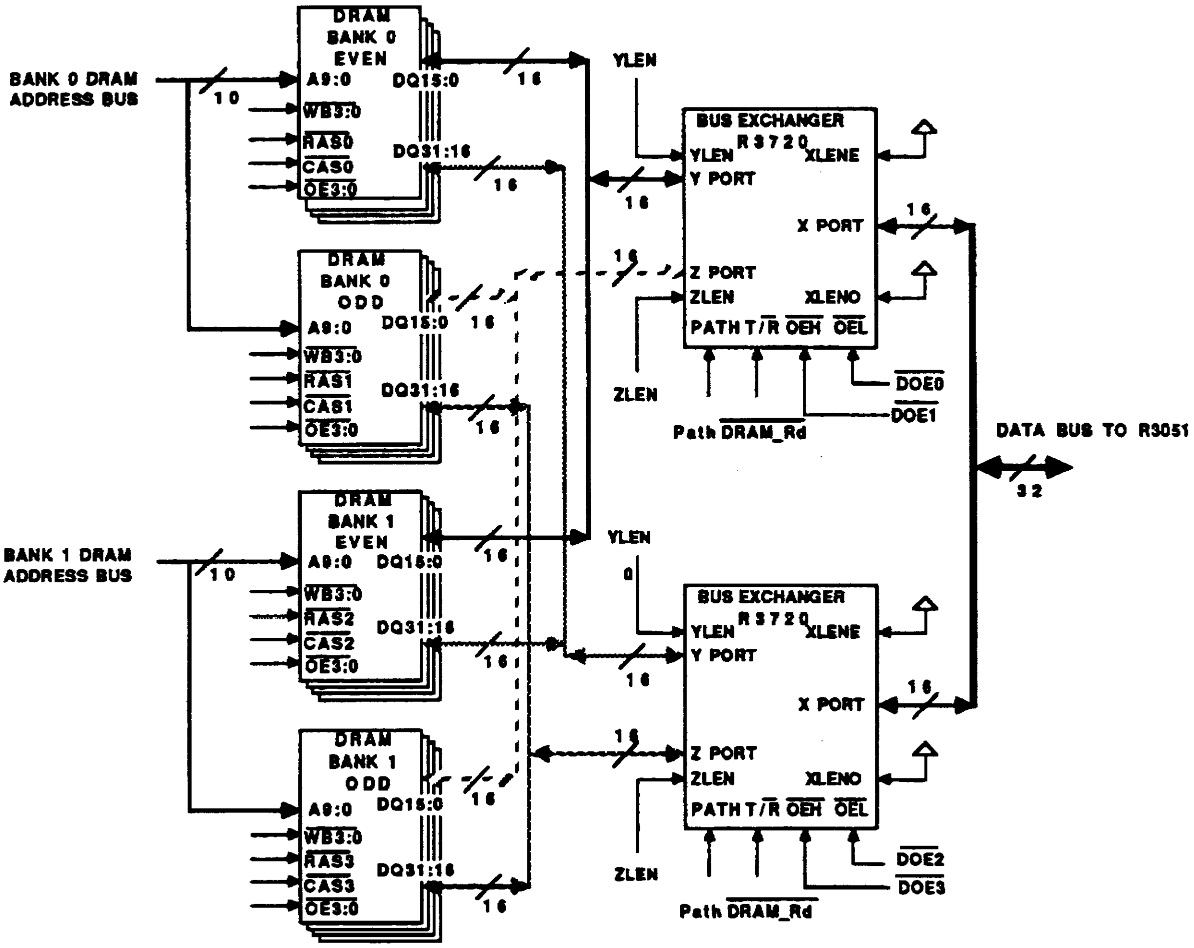
ADDRESS MULTIPLEXING & MEMORY DRIVERS

OTHER SCHEMES ARE POSSIBLE ALSO.

MAKE SURE ADDRESS SETUP & HOLD TIMINGS ARE MET

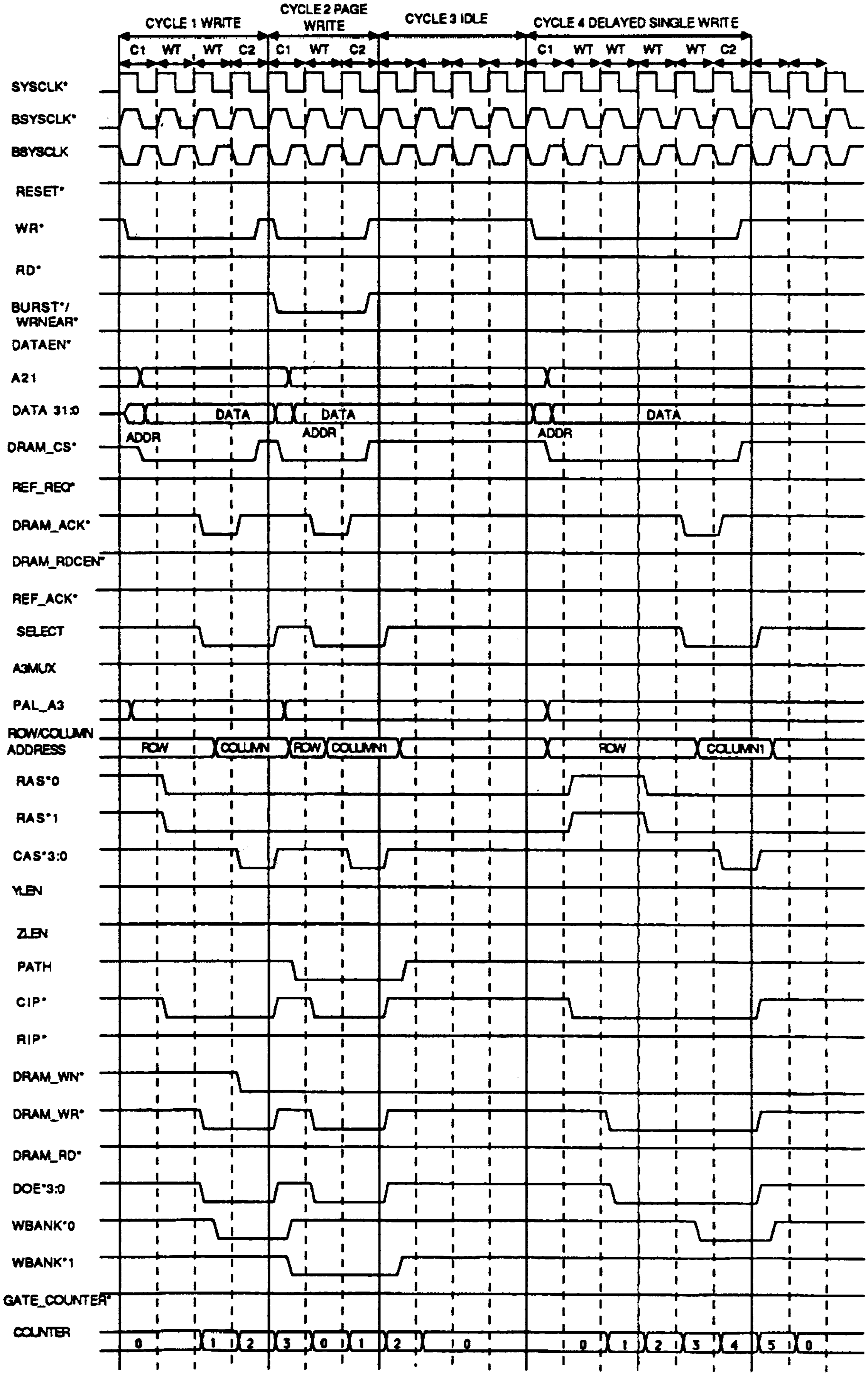


DRAM STATE MACHINE IMPLEMENTED IN DISCRETE LOGIC. VERY FAST PALS ARE USED FOR 33 MHz & ALMOST OPTIMUM PERFORMANCE. SLOWER DEVICES CAN BE USED AT THE EXPENSE OF A CLOCK CYCLE FOR 25MH LOWER DEVICES ARE POSSIBLE THE DESIGN CAN BE IMPLEMENTED



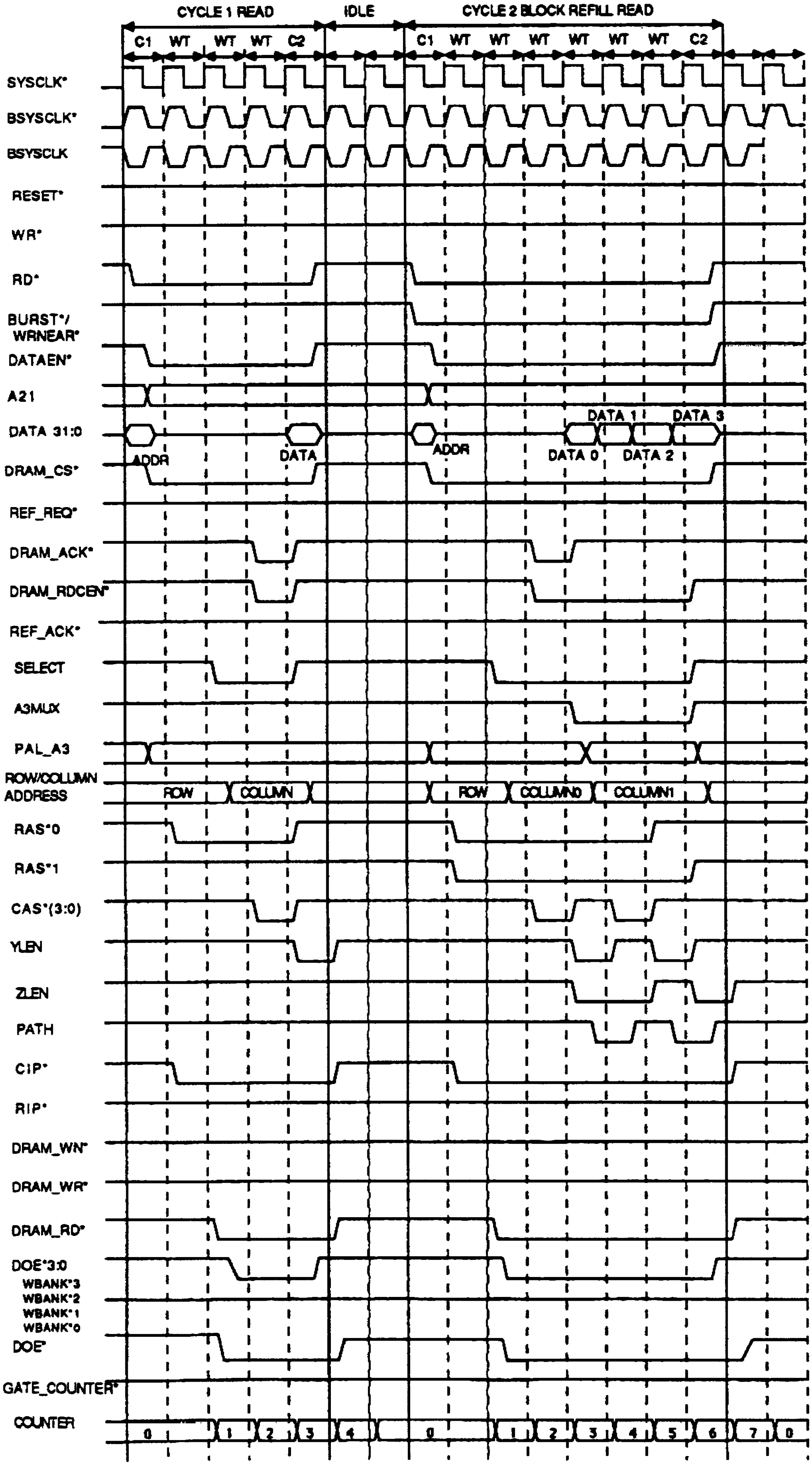
DATA PATH & DRAM STRUCTURE.

VALID FOR ANY INTERLEAVED DESIGN.



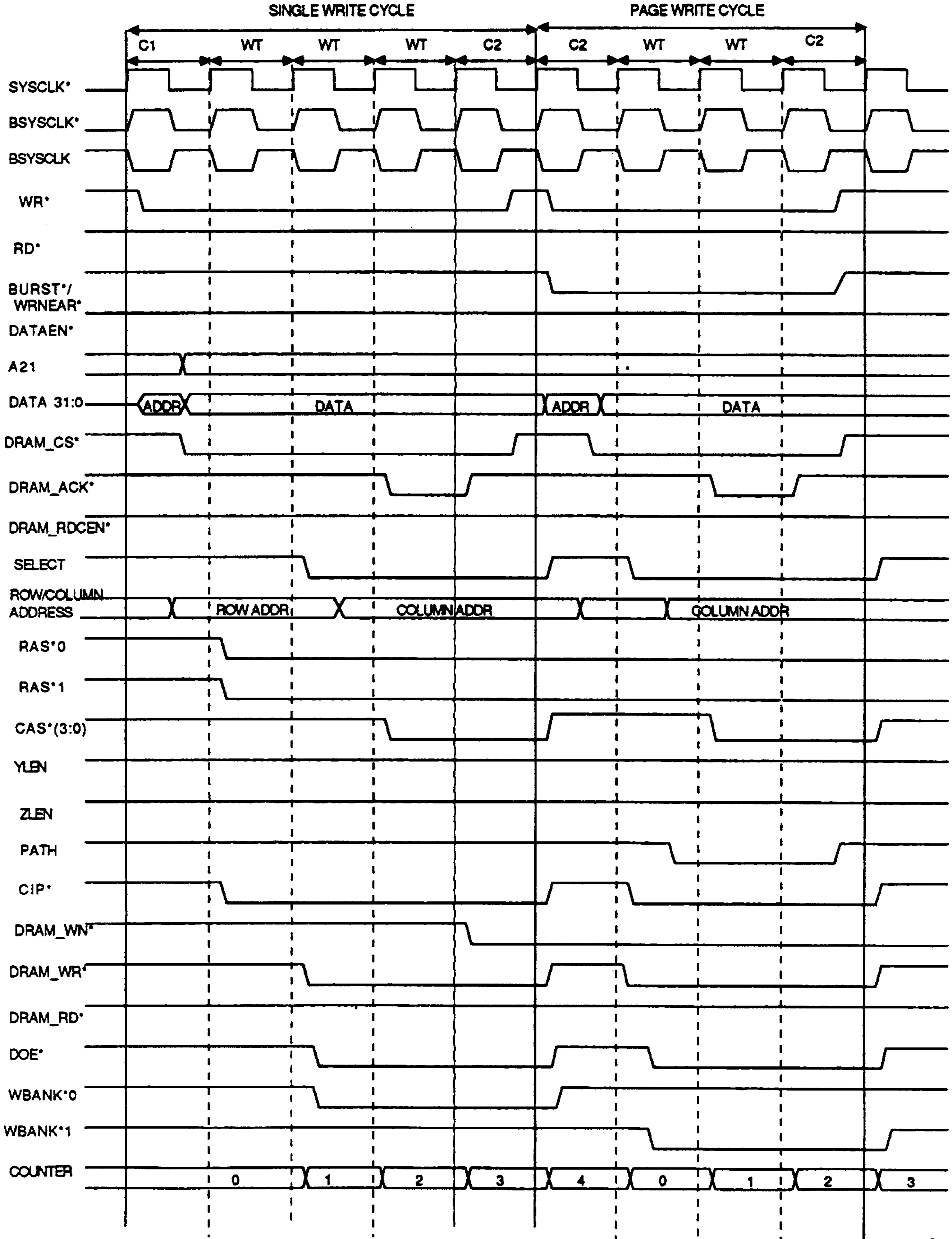
LMOST OPTIMUM TIMING PERFORMANCE AT 33 MHZ & 25 MHZ USING $\overline{\text{SYSCLK}}$ STATE MACHINE.

7

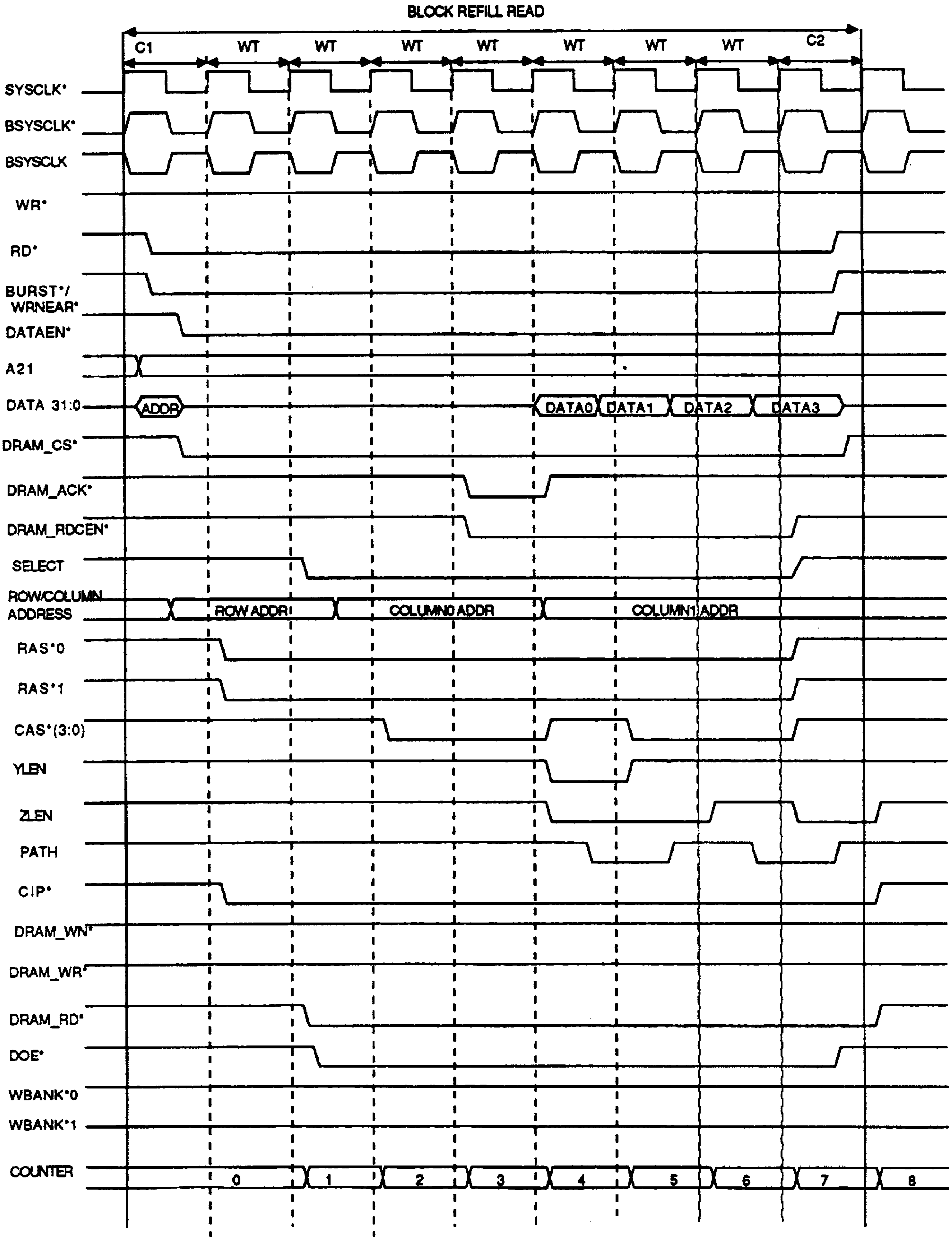


SAME AS PREVIOUS

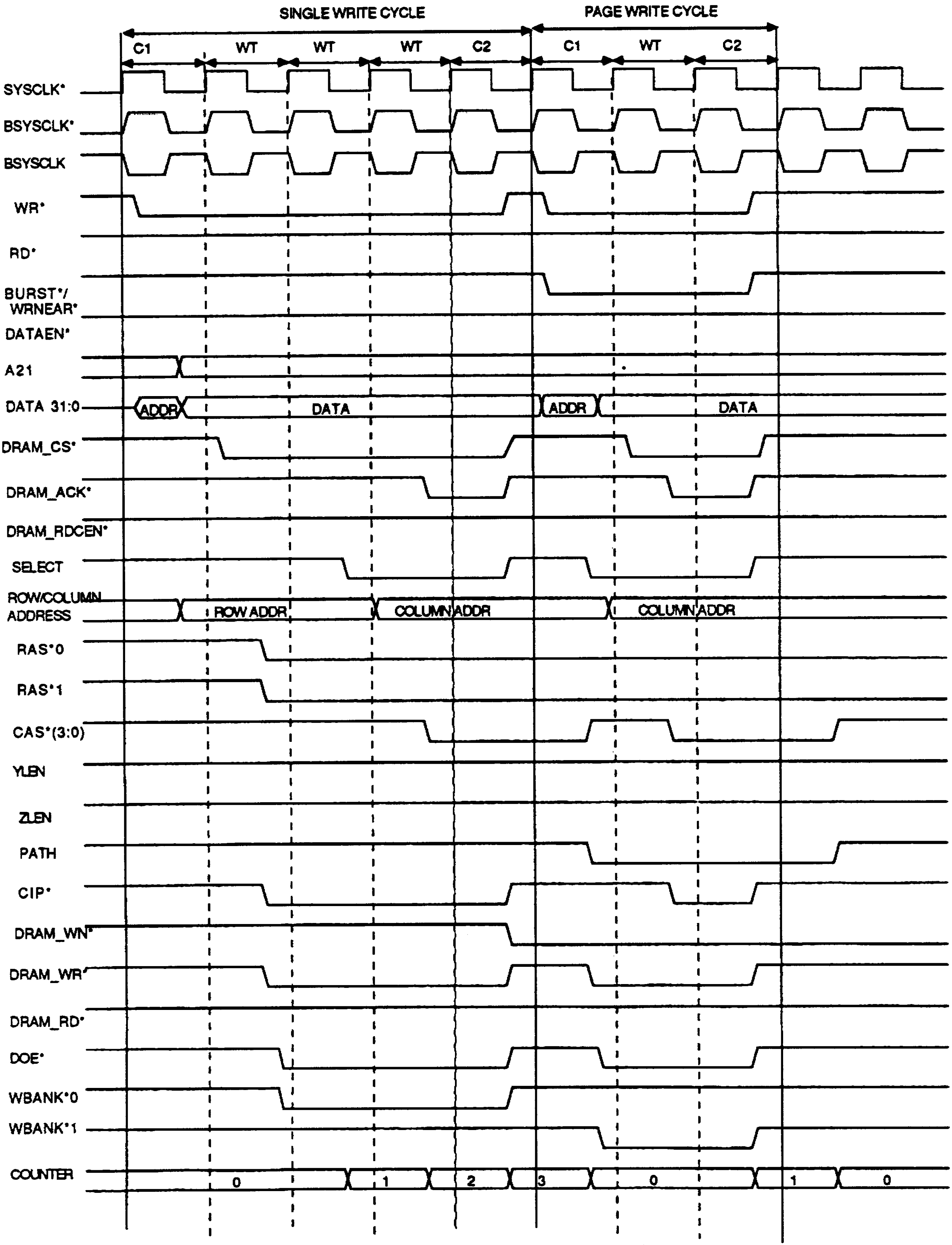
OCT 21 '92 08:55AM IDT RISC MPR P.8/12



LESS OPTIMUM TIMING PERFORMANCE AT 33MHz BY USING SLOWER DEVICES & SACRIFYING A CLOCK CYCLE. STILL USING SYSCLK STATE MACHINE.

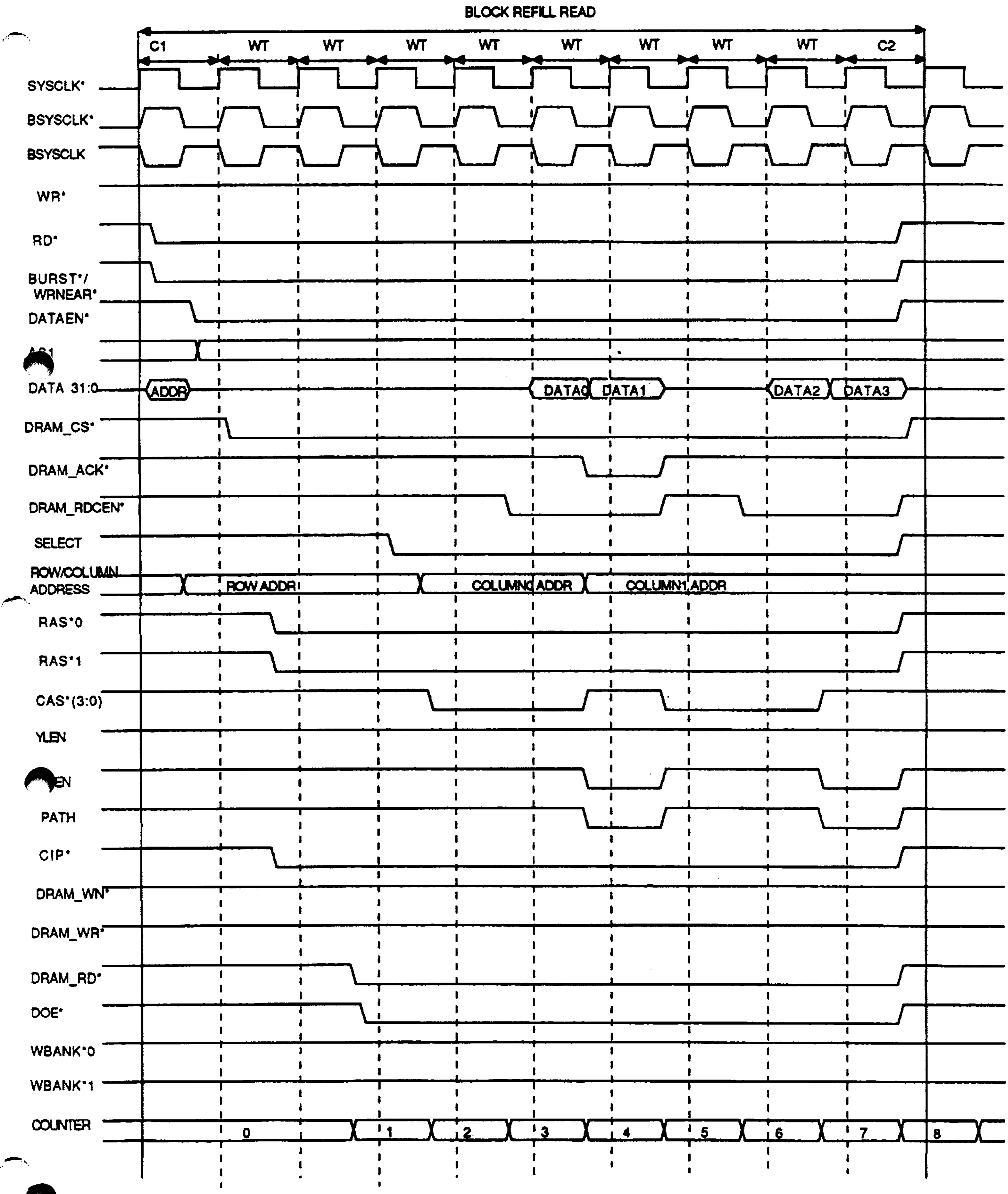


SAME AS PREVIOUS.



LESS OPTIMUM TIMING PERFORMANCE AT 33 MHz BY USING
SLOWER DEVICES & SACRIFYING ONE CLOCK CYCLE.
USING SYSCLK STATE MACHINE (INVERTED SYSCLK).

(10)



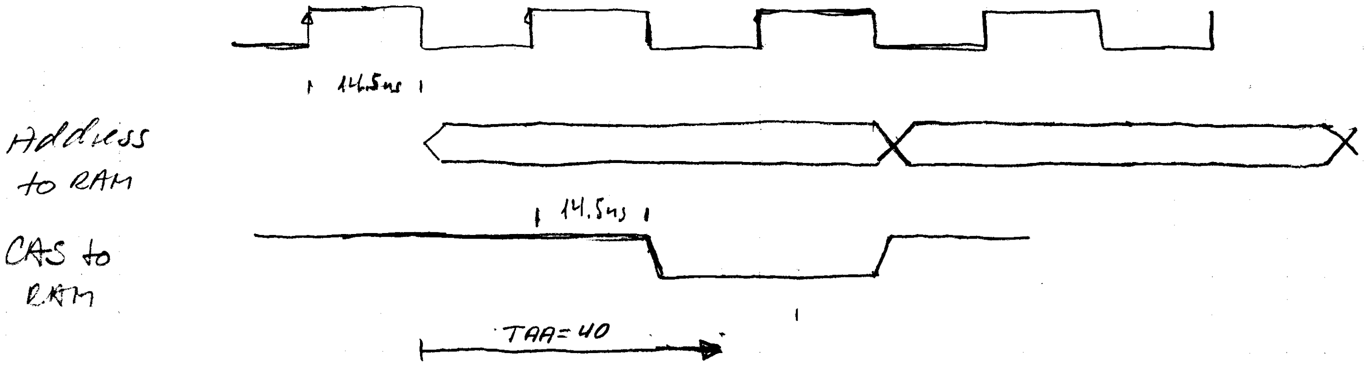
SAME AS PREVIOUS.

KAN

24/11-92

Styring af adresselatch

Address even



Adressen til even latches ikke.

setuptime > 15ns

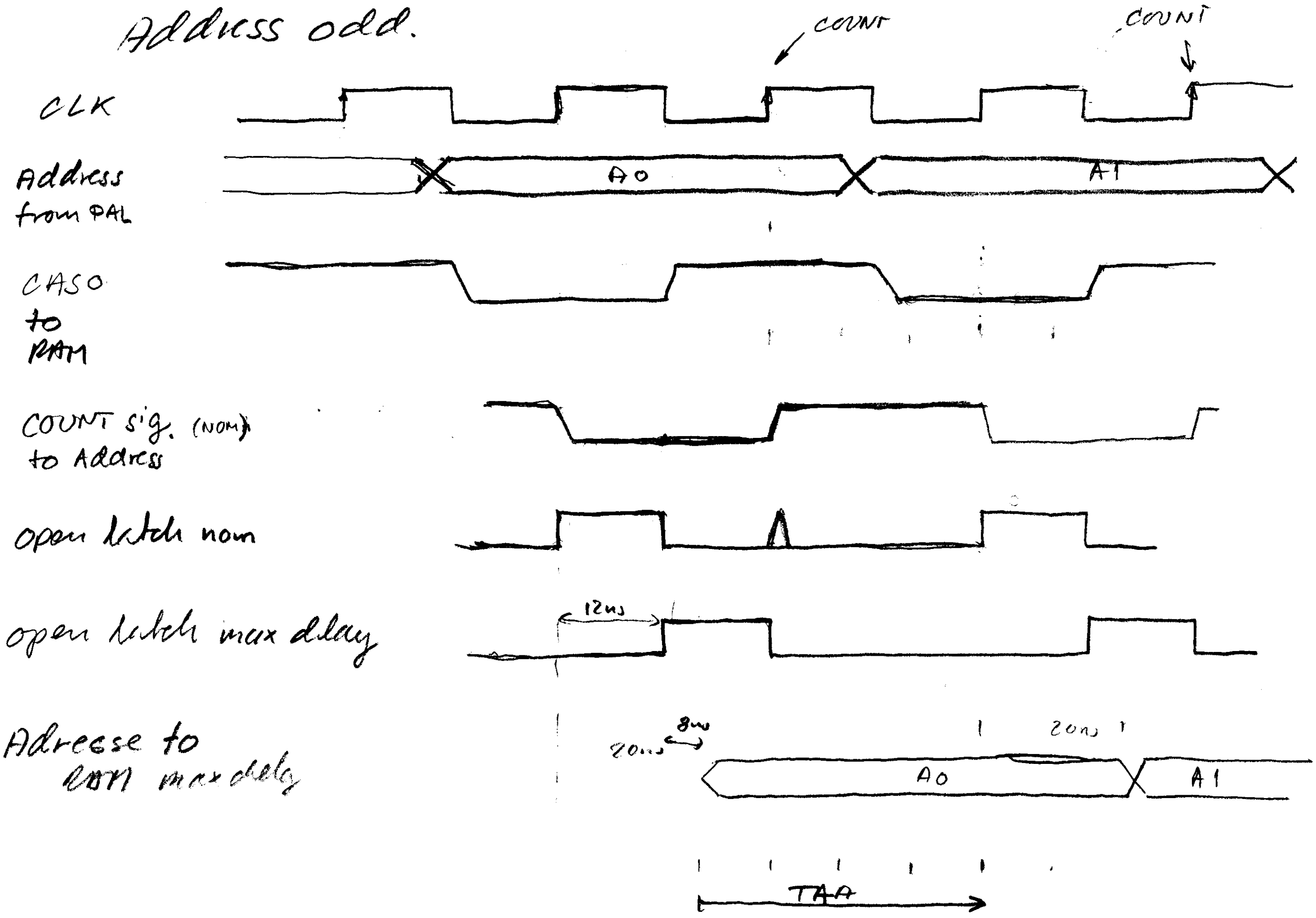
holdtime > 15ns

access time fram address TAA er ikke en begrænsende parameter, når delay fra klok til CASE er større end 14.5 ns og det er den.

KMN
24/11-92

Styring af adresselatch

Address odd.



Adresselatch åbnes i første klokperiode, når der er count

address hold time >15ns

address set up time: address min 10 ns før klok som sætter CAS.

hvis klok to CAS delay < 10ns, er address access time begrænsende.

Styring af adresselatch

Ved beregningerne er LO lavet med et kombinatorisk output d.v.s delay fra klok til output på 12 ns. Det ville være ønskeligt at anvende et klokket output med max delay på 8 ns.

Generelt :

I hviletilstand skal latchen være åben

$\overline{n2}+\overline{n1}$.

I øvrige tilstande er latch lukket med mindre adressen tælles op FCTO = 1

$$LO = \overline{n2}+\overline{n1} + FCTO \cdot CLK$$

KAN 25/11-92

Skrivninger

CAS generes af MACH 210-12, som driver 74FBT2841B.

max delay fra klok til CAS : PAL : 8 ns

DRIVER 6.5 ns

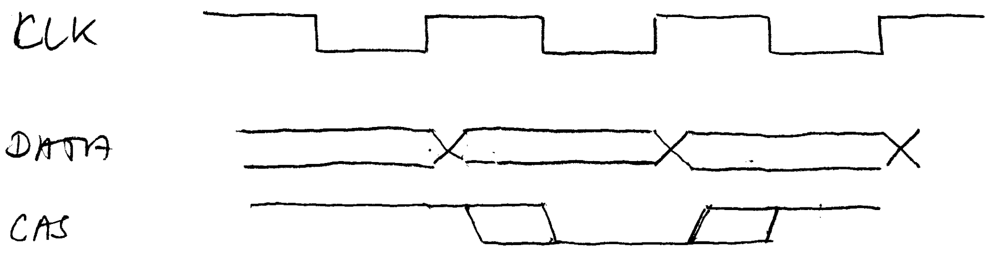
belastning på CAS : min 4 RAM 14.5 ns.

i datavejen findes FCT 16500AT

max delay fra klok til data : 5.6 ns.

max belastning 2 RAM'

Disse registre klokkes på hver klok:



RAM kræver : data setup 0 ns

data hold 15 ns

Data hold time er altid opfyldt.

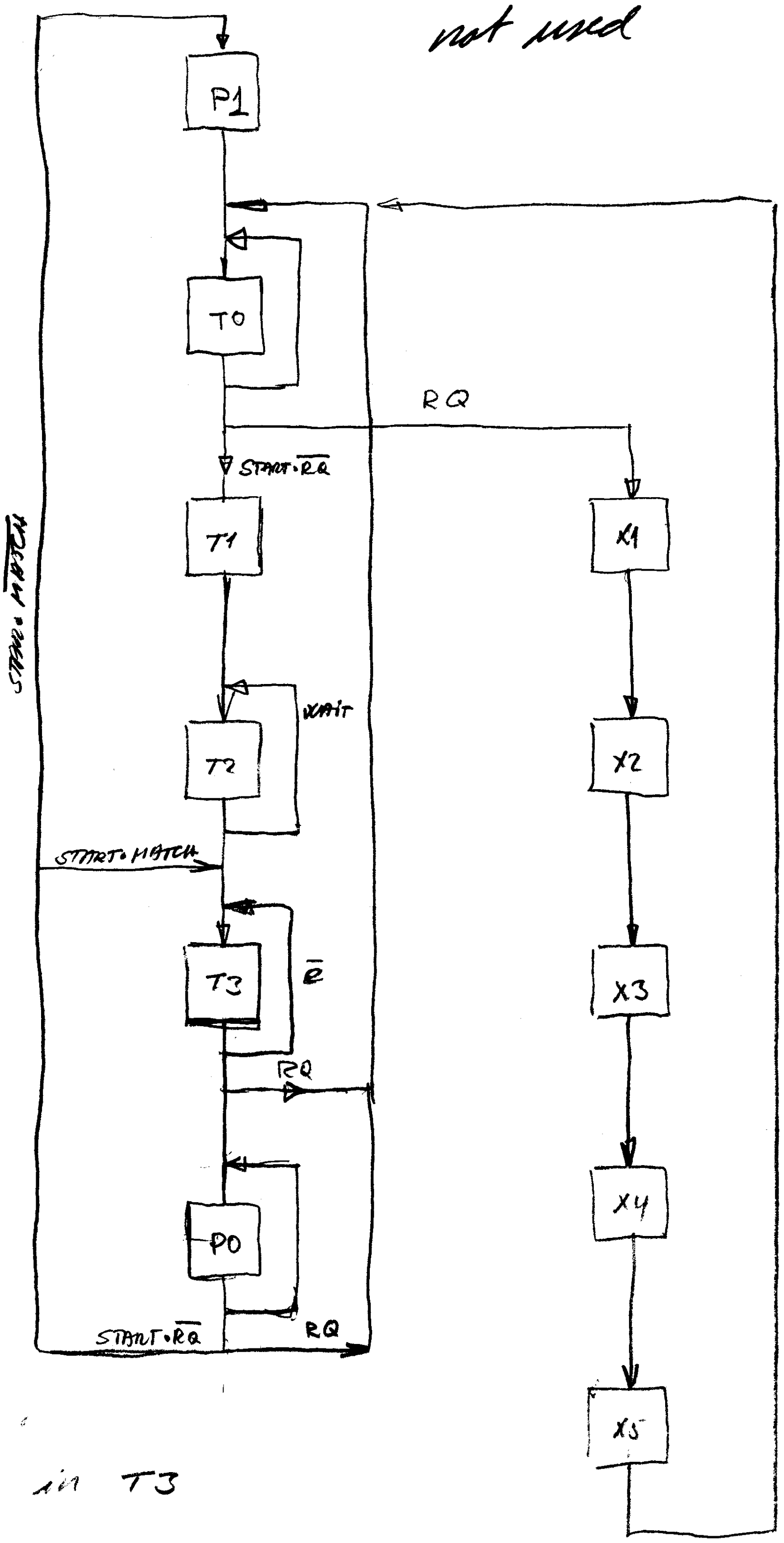
Data set up time er opfyldt, når delay i

PAL + 74FBT2841B > delay i FCT 16500AT

DROMB state diagram, reduced

KON
3/11-92

not used



CBS generation in T3

4 state variable

CAS generation READ 11/11-92 KTN

finder sted i tilstand T3. Realiseres med 3 slavemaskiner CMSE, CMSO, OE. Slavemaskinerne kommunikere indbyrdes v.h.a variable FO, FE. maskinerne holdes i inaktiv tilstand, når hovedmaskinen ikke er i T3.

Maskinernes start kan beskrives på 2 måder:

A. Når hovedmaskinen hopper til T3 sætter den alle variabele til slavemaskinerne, hvorved maskinerne starter i tilstande bestemt af hovedmaskinen.

B. maskinerne har samme input som hovedmaskinen, og hopper selv til den rigtige tilstand, når hovedmaskinen hopper til T3 $\left\{ \begin{array}{l} \text{T2}\cdot\overline{\text{WAIT}} \\ \text{PO.MATCH.START.RQ} \end{array} \right.$

Bemærk: det er kun en forskel i beskrivelsen resultatet bliver det samme. Metode B anvendes da den er mere overskuelig.

<u>input, som giver start</u>

1. tilstandsvariable fra hovedmaskine plus WAIT, MATCH, START RQ

2. <u>input, som giver slutbetingelser</u>

   AL: address low, BURST, READ.

3. interne variable: FO, FE $\left\{ \begin{array}{l} \text{full odd} \\ \text{full even} \end{array} \right.$

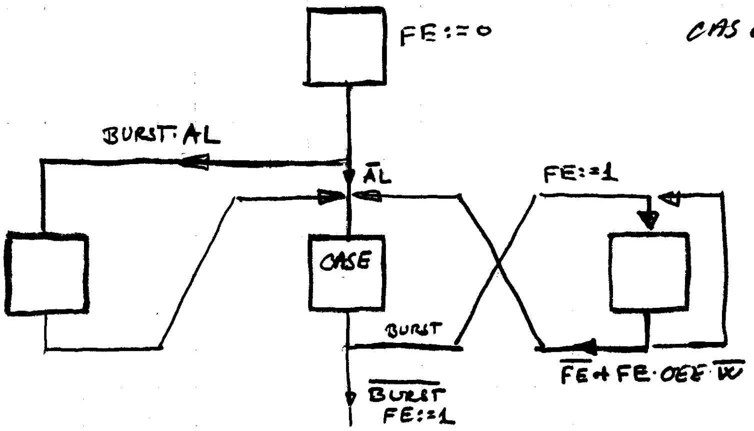
4. slutbetingelser

   Burst: intet problem: 1 cycle.

   BURST $\left\{ \begin{array}{l} \text{BURSTC 4 tællinger} \\ \text{BURSTOM: indtil CS inaktiv.} \end{array} \right\}$ samles i et signal END.

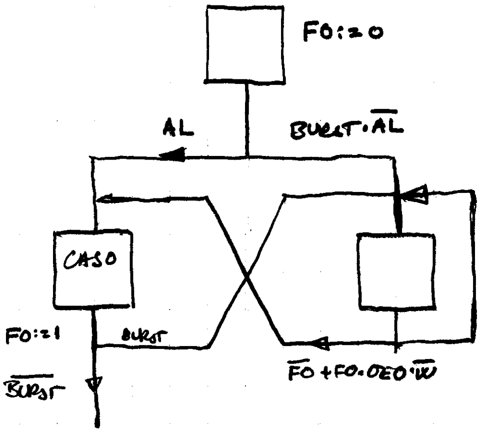
KAN 11/11-92

State diagrams: READ CAS generation



CAS even : CASE

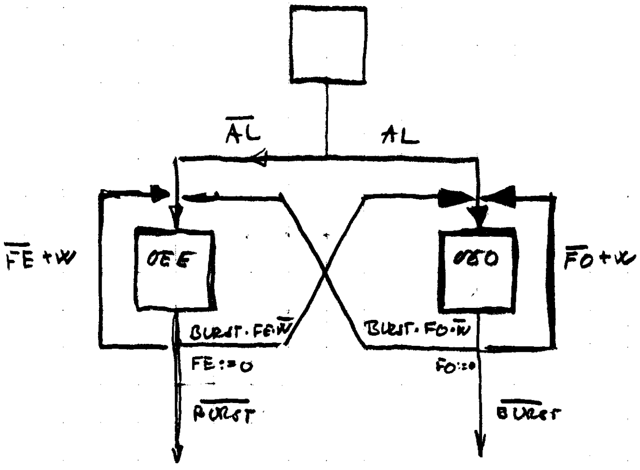
2 state variable



CAS odd CASO

1 state var

output enable



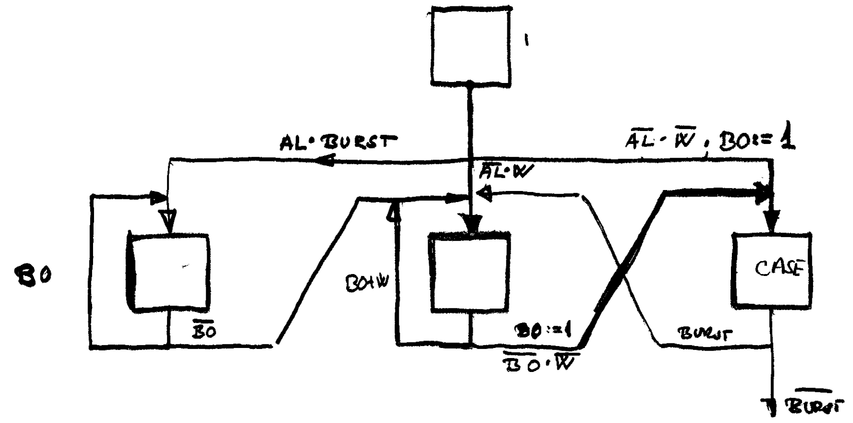
0 state var

state FE,FO

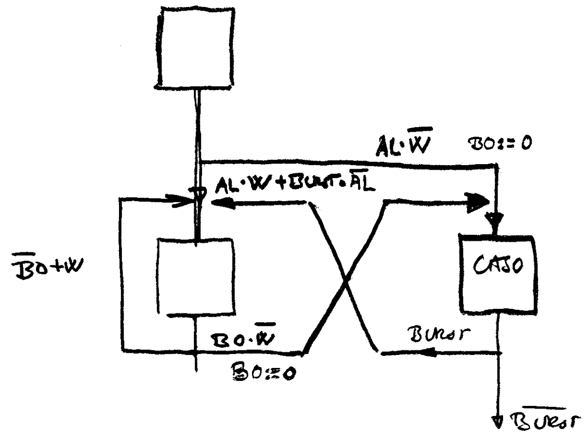
initial variable : 3 + 2 = 5

Statediagrams WRITE CAS generation

CAS even



2 var



1 var

ved udhop sattes BO på følgende måde uafhængigt af Burst

W \ AL	0	1
0	1	0
1	0	1

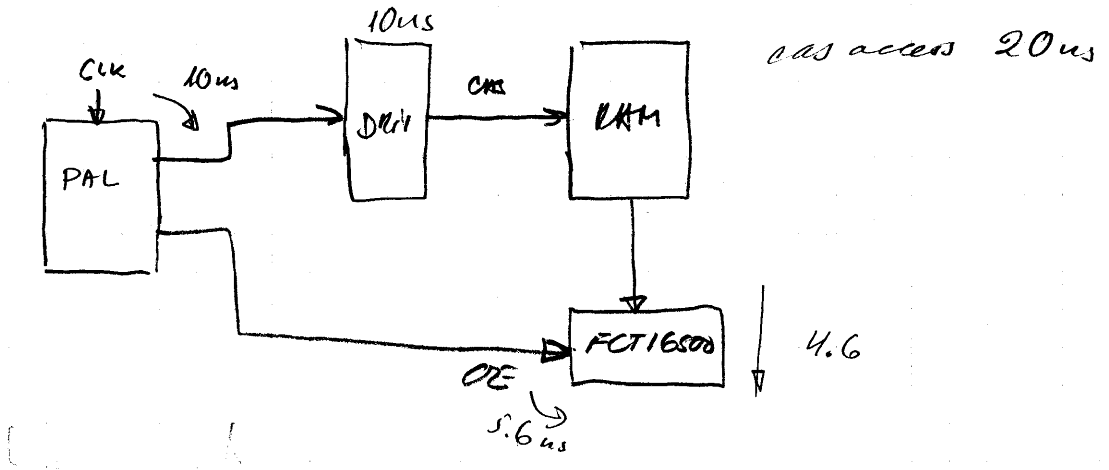
state BO

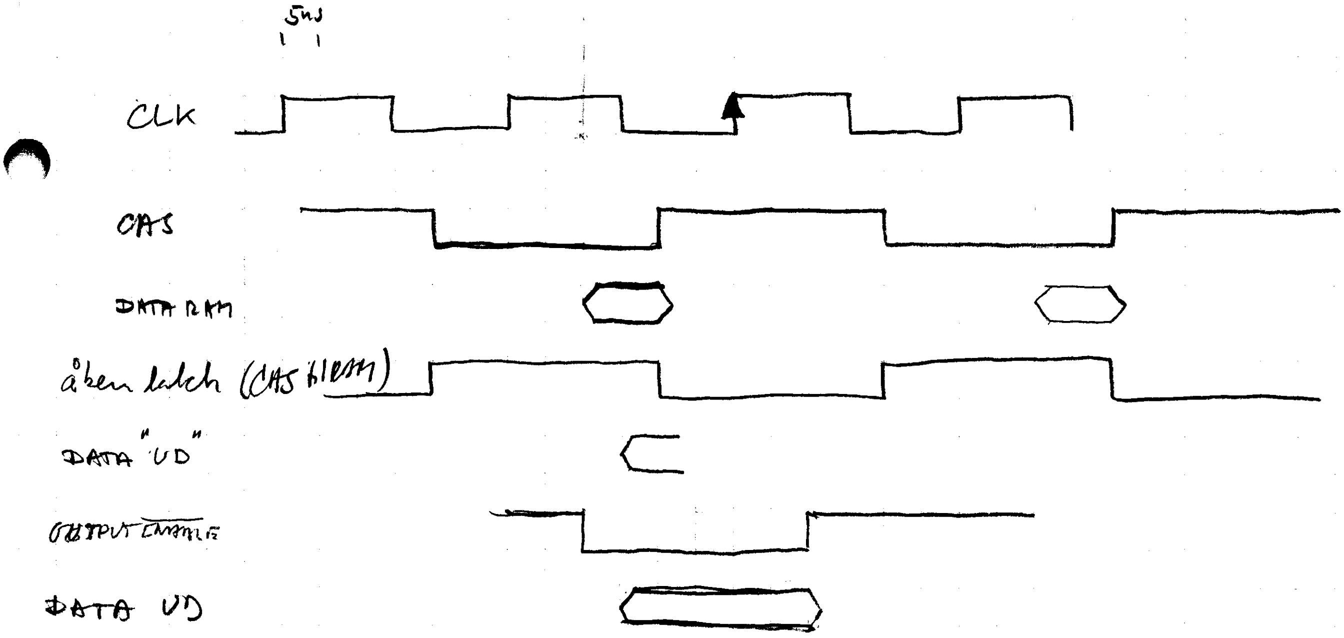
antal variable: 3 + 1

KJN 21/10-92

Ram control

READ





Data set up til latch min 15.4ns.

bestemmes af:	PAL latch til output :	10	8
	output enable 16500 :	5.6	5.6
		15.6	rest 14.4
		13.6	rest 16.4

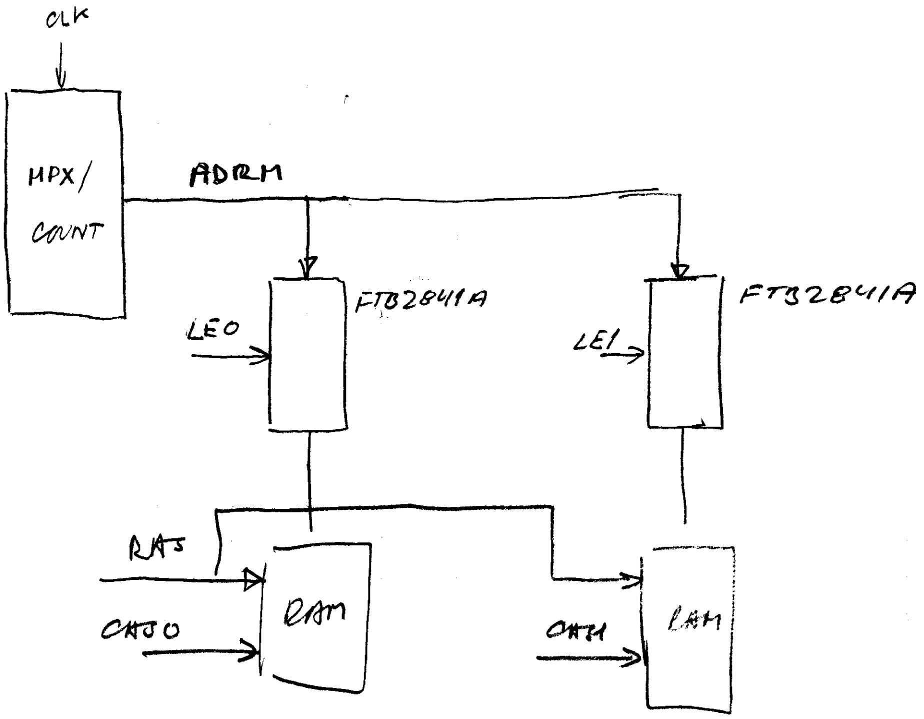
beregning gælder, når driver er hurtigere end 10 ns.

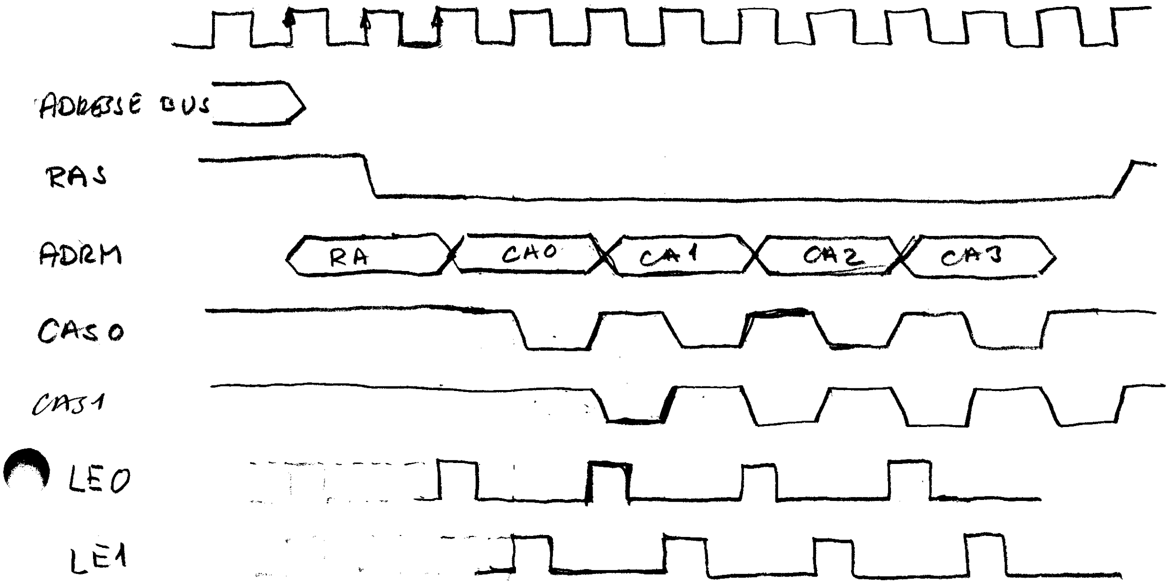
hvis driver langsommere gælder:

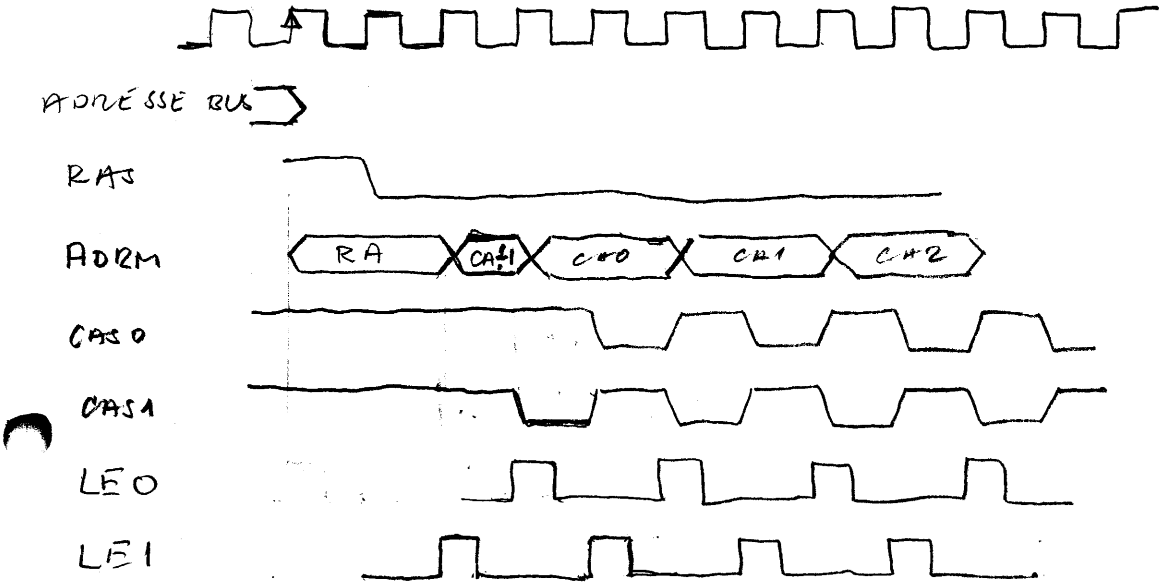
Pal	latch to out	10	15
	driver	10	
	cas access	20	20
	16500 gennemløb	4.6 ~	5
		44.6	40

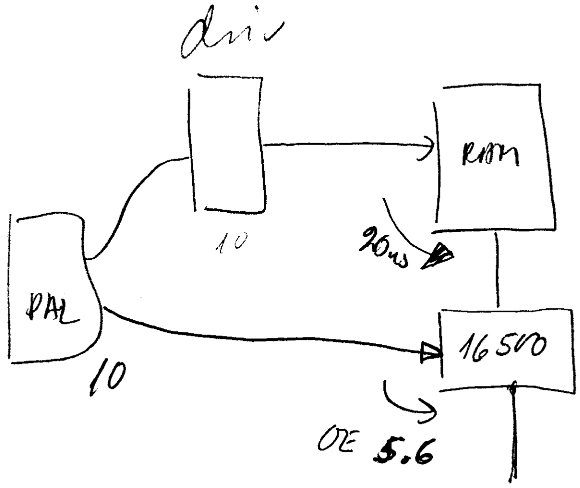
Rest 60-44.6 = 15.4
20.ns

Adresse

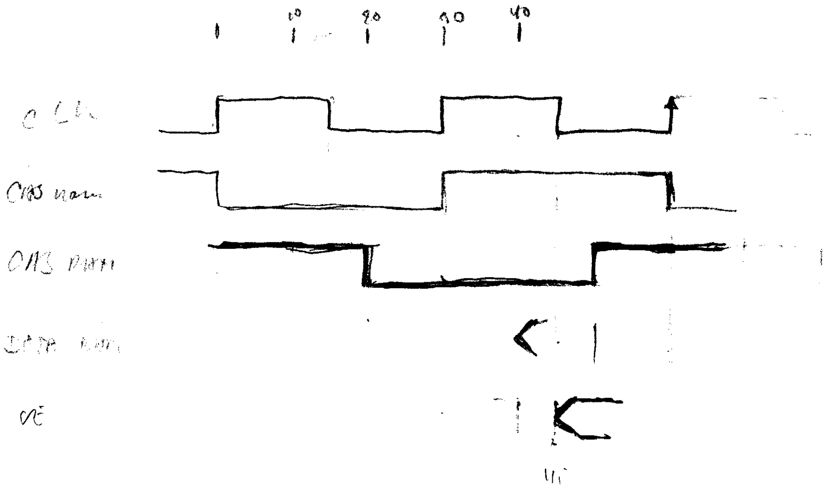

CLK
MPX/
COUNT
ADRM
FTB2841A
FTB2841A
LE0
LE1
RAS
CAS0
CAS1
RAM
RAM


ADRESSE BUS
RAS
ADRM
RA
CA0
CA1
CA2
CA3
CAS0
CAS1
LE0
LE1


ADRESSE BUS
RAS
ADRM
RA
CA1
CA0
CA1
CA2
CAS0
CAS1
LE0
LE1



Clock to output max 5.3 setup 1.5



KMN 26/10-92

Lager cycles

Refresh

Read single

Write single

Read burstc { 4 ords burst

Read DMA

Write DMA

Alle cycles starter på START m.m.

Single cycles afsluttes når cycle er udført

Read burstc afsluttes af intern tæller

Read DMA : afsluttes af DMA burst.

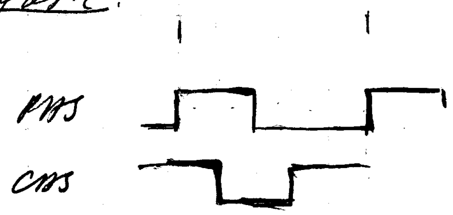
Write DMA : afsluttes af DMA burst.

Refresh i DMA burst?

Lager kan forsinkes med signalet wait.

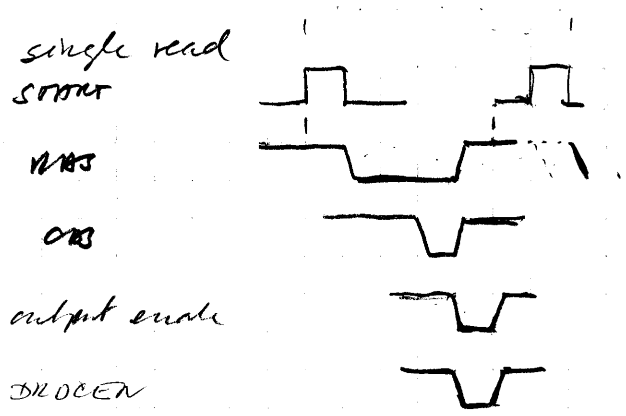
lager cycles

<u>Refresh:</u>

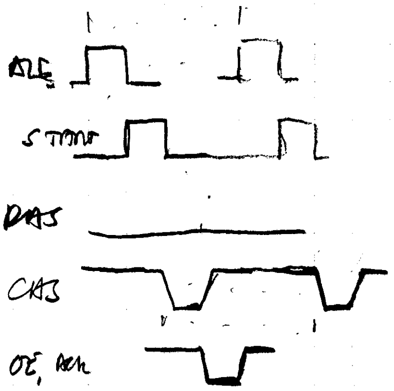


6 180ns

single read

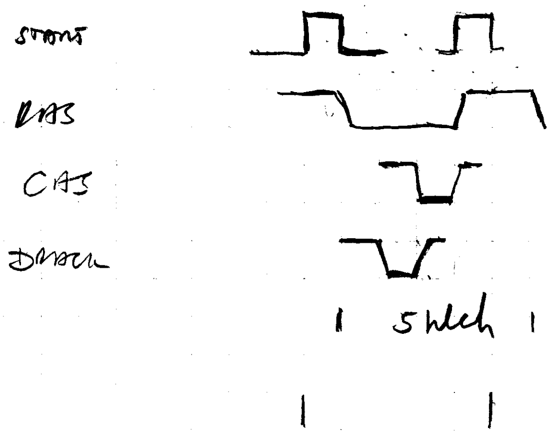


4 klok PAGE MODE ~ 120 ns

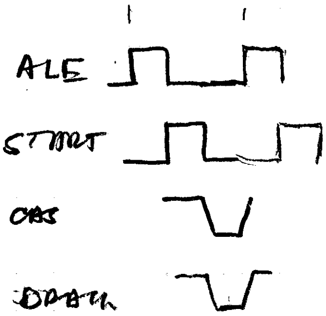


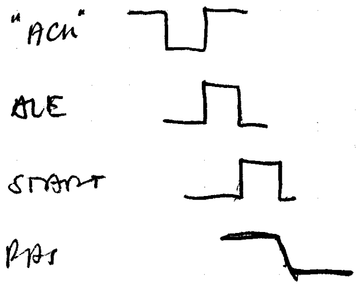
4 klok

single write



3 klok





dde Dansk Data Elektronik A/S

Initials KAN	Page 1
Date 27/11-92	Project

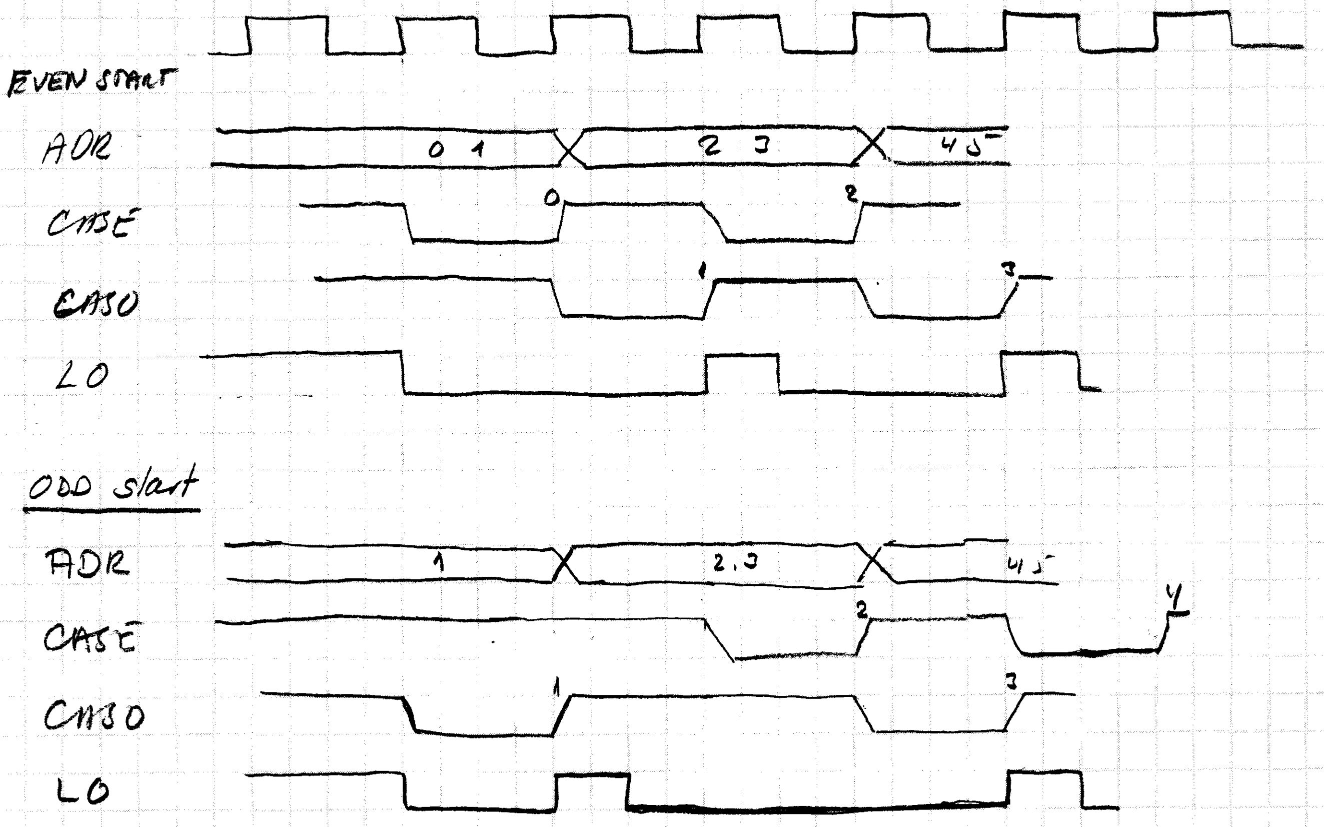
styring af adresselatch til odd bank.

I hviletilstande er latch åben. I alle tilstande navngivet C.. åbnes latch af signal.

Ved lige startadresse tælles adressen op efter CASE. Ved ulige startadresse tælles adressen op efter første CASO og derefter efter CASE.

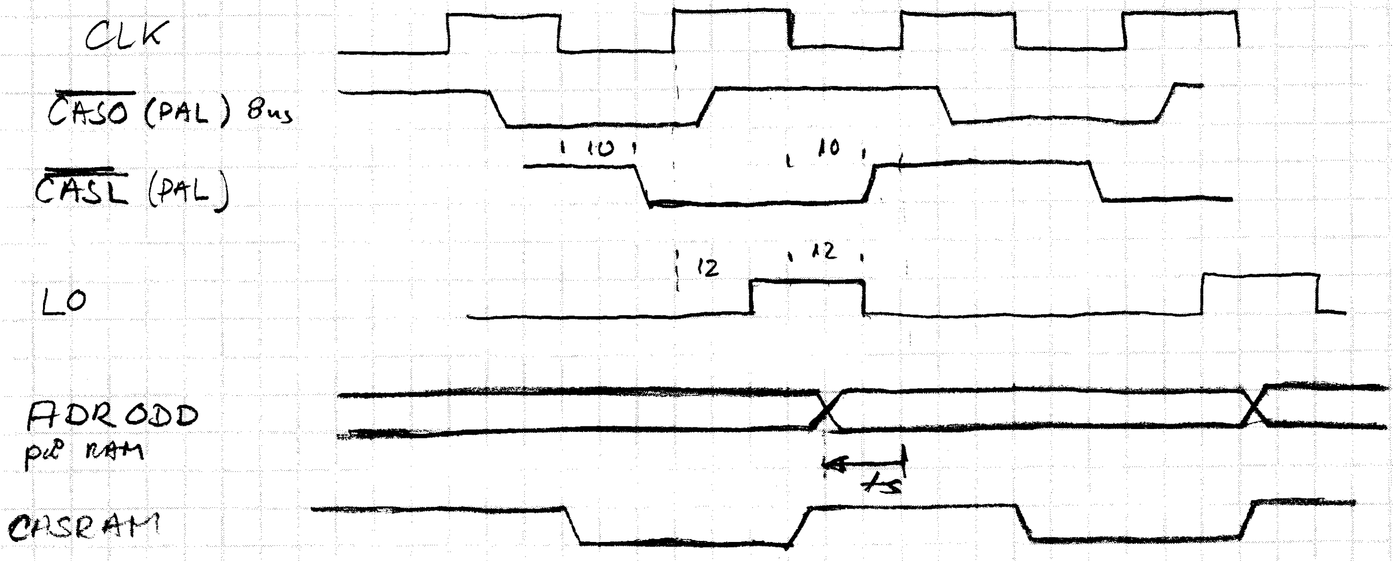
Adresselatch åbnes i første halvperiode efter CASO.

Nominel timing:



Styring af adresse latch

CAS0 latches i DRAMB og gates sammen med klok.

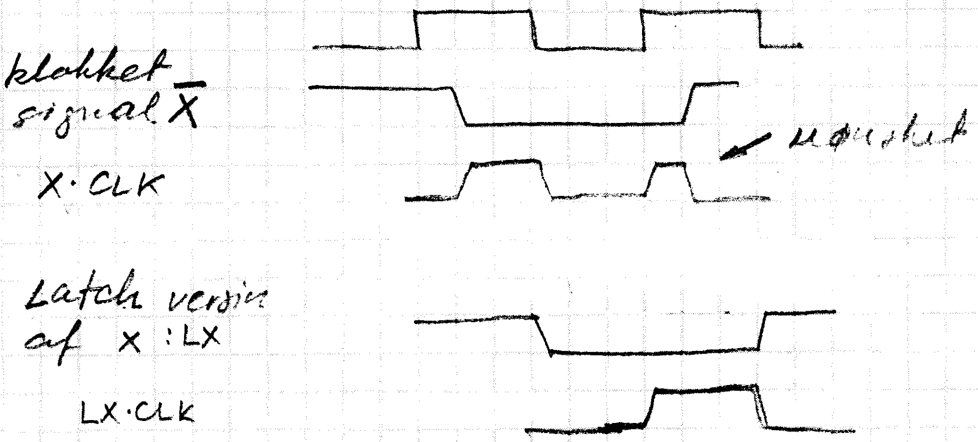


address set uptime to clock ↑ : min 10ns

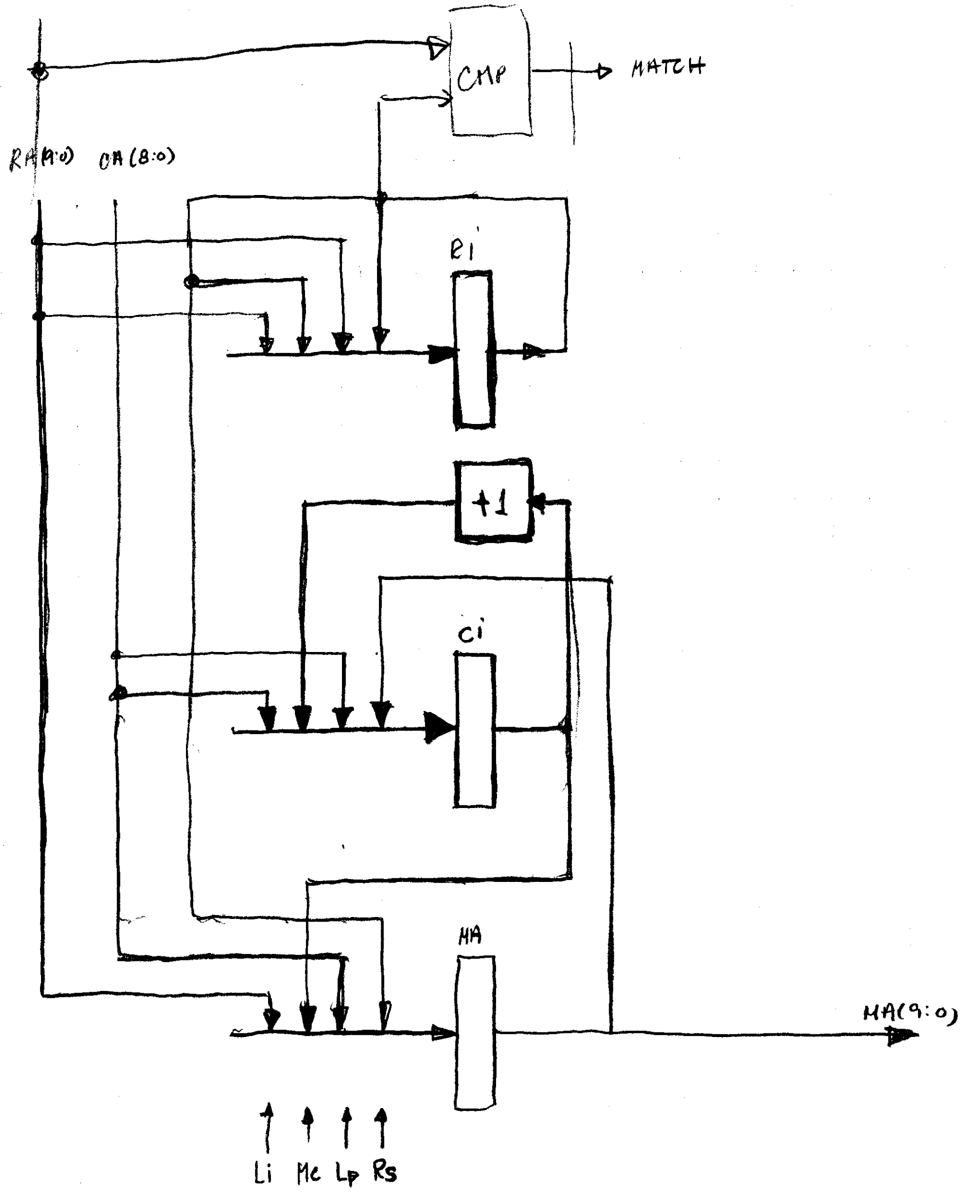
clock cycle	30
komb PAL	12
latch enable	8
setuptime	10 ns

address hold time from CAS↓ > 15 ns

anvendelse af latched version af CAS



DRAMA



Li: Load idle

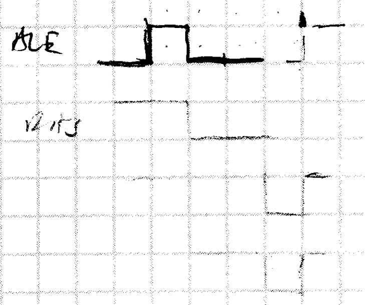
Mc: mux/count

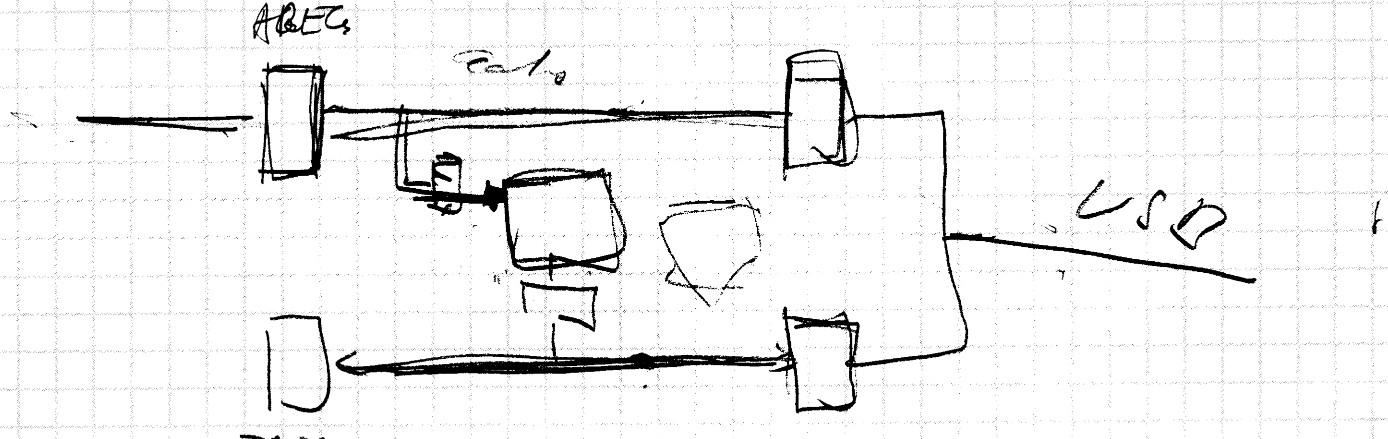
Lp: Load page

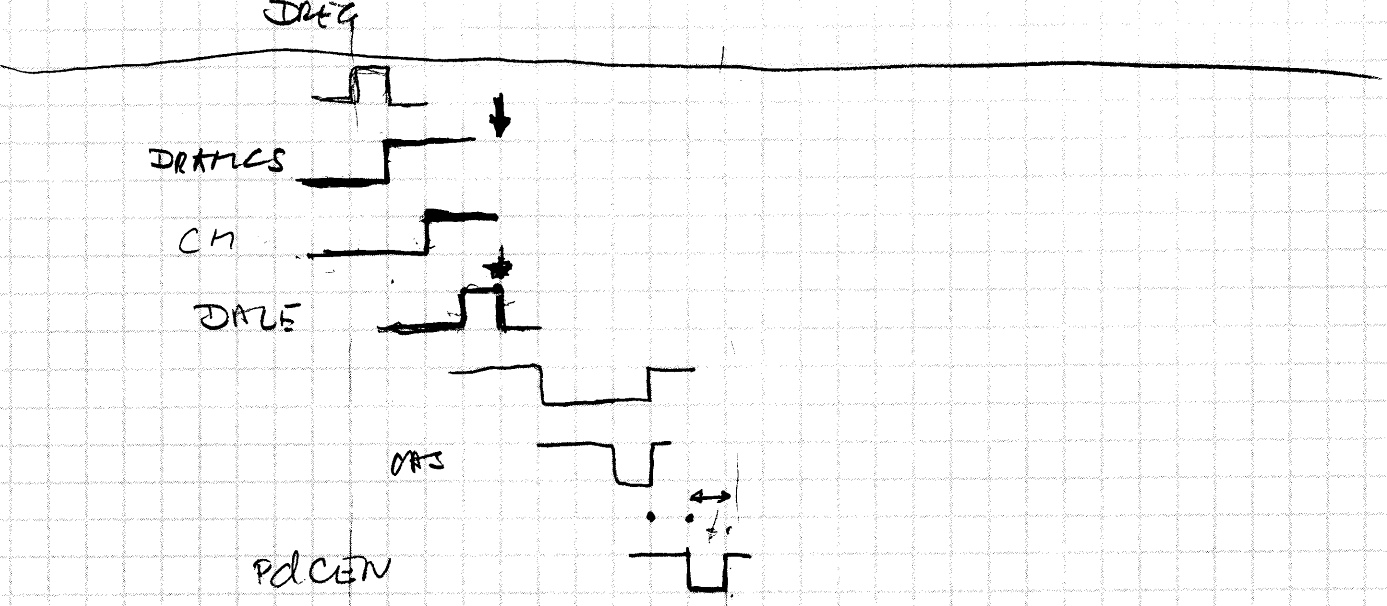
Rs: Restart

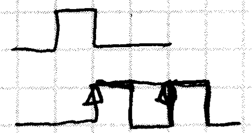
dde Dansk Data Elektronik A/S

	Initials	Page
	Date	Project









```
"Dram controller   DRAMA
"Address multiplexer, comparator, and counter.          kan 921120
"8 bit cas address counter.

module drama;

declarations;

"Address inputs:

"ra(9:0):       ras address, most significant address bits.
"ca(8:0):       cas address, least significant address bits.

ra9, ra8, ra7, ra6, ra5, ra4, ra3, ra2, ra1, ra0        pin;
     ca8, ca7, ca6, ca5, ca4, ca3, ca2, ca1, ca0        pin;

"Address outputs:

"ma(9:0):       multiplexed  address bits to DRAM-array.

ma0     pin  4 istype 'reg_d,buffer'; "A4
ma1     pin  6 istype 'reg_d,buffer'; "A8
ma2     pin  8 istype 'reg_d,buffer'; "A12

ma3     pin 19 istype 'reg_d,buffer'; "B4
ma4     pin 17 istype 'reg_d,buffer'; "B8
ma5     pin 15 istype 'reg_d,buffer'; "B12

ma6     pin 24 istype 'reg_d,buffer'; "C4
ma7     pin 30 istype 'reg_d,buffer'; "C12

ma8     pin 43 istype 'reg_d,buffer'; "D4
ma9     pin 37 istype 'reg_d,buffer'; "D12


"Internal registers:

"ri(9:0):       holds the current page (ras) address.
"               The content of the register is compared with
"               ra(9:0) to determine if the access is within
"               the current page.

ri0     node 46 istype 'reg_d,buffer'; "A3
ri1     node 48 istype 'reg_d,buffer'; "A7
ri2     node 52 istype 'reg_d,buffer'; "A15

ri3     node 54 istype 'reg_d,buffer'; "B3
ri4     node 56 istype 'reg_d,buffer'; "B7
ri5     node 60 istype 'reg_d,buffer'; "B15

ri6     node 61 istype 'reg_d,buffer'; "C1
ri7     node 68 istype 'reg_d,buffer'; "C15

ri8     node 69 istype 'reg_d,buffer'; "D1
ri9     node 76 istype 'reg_d,buffer'; "D15
```

```
"ci(9:0):       cas address counter. increments the cas address
"               during page mode accesses. Implemented with trigger
"               flip flops. Also used for storing RAS addresses.
"               bit 9 and 8 are not part of the counter.

ci0                     node 45 istype 'reg_t,buffer'; "A1
ci1                     node 47 istype 'reg_t,buffer'; "A5
ci2                     node 50 istype 'reg_t,buffer'; "A11

ci3                     node 53 istype 'reg_t,buffer'; "B1
ci4                     node 55 istype 'reg_t,buffer'; "B5
ci5                     node 58 istype 'reg_t,buffer'; "B11

ci6                     node 63 istype 'reg_t,buffer'; "C5
ci7                     node 67 istype 'reg_t,buffer'; "C13

ci8                     node 71 istype 'reg_t,buffer'; "D5
ci9                     node 75 istype 'reg_t,buffer'; "D13

cc                      node    istype 'reg_d,buffer';

"Control inputs.

"ale:           Address latch enable. The address must be stored on
"               this signal. The cycle may or may not be a memory
"               access. This is determined by DRAMB.

ale                             pin;

"fct(1:0):      Function code from DRAMB. Controls the function of
"               DRAMA.

fct1, fct0                      pin;

"clk:           Clock input. Inverted version af !SysClk.

clk                             pin;

"Control outputs.

"match(3:0):    Output signals to DRAMB. Give the results of the
"               comparison between ra and ri.
"               match0 compares bit 1,0
"               match1 compares bit 4,3
"               match2 compares bit 7,6,2
"               match3 compares bit 9,8,5
"

match0                  pin  9 istype 'reg_d,buffer'; "A14
match1                  pin 14 istype 'reg_d,buffer'; "B14
match2                  pin 28 istype 'reg_d,buffer'; "C8
match3                  pin 39 istype 'reg_d,buffer'; "D8

"start:         Output signal to DRAMB. Indicates potential
"               memory cycle.
```

```
start                           pin istype 'reg_d,buffer';

"NOTICE:        ALL PINS ARE USED.

"               20 of 32 burried registers are used

"#######################################################################
"                      Description of function.
"#######################################################################

"Load idle:  li

"This function is used when the DRAM controller is idle and not in page
"mode.

"ma := ra;      ci := ca;       ri := ra;

"ras addr       cas addr        ras addr

"Multiplex/count: mc

"This function is used to multiplex and count the address.

"ma := ci;      ci := ci+1;     ri := ri;

"cas addr       cas addr + 1    ras addr


"Load page mode:  lp

"This function is used when the DRAM controller is idle and
"in page mode.  ri is compared to ra. match is set accordingly.

"ma := ca;      ci := ca;       ri := ra;

"cas addr       cas addr        ras addr

"On the first clock after ale: ci:= ci+1;

"Restart:  rs

"This function is used after no match in page mode.

"ma := ri;      ci := ma;       ri := ri

"ras addr       cas addr        ras address

"Hold: hl

"do nothing.

c,z,x = .C., .Z., .X.;

fct = [ale, fct1, fct0];
```

```
equations;

"#######################################################################
"                       definition of clocks
"#######################################################################

ma9.c = clk; ma8.c = clk; ma7.c = clk; ma6.c = clk;  ma5.c = clk;
ma4.c = clk; ma3.c = clk; ma2.c = clk; ma1.c = clk;  ma0.c = clk;

ri9.c = clk; ri8.c = clk; ri7.c = clk; ri6.c = clk;  ri5.c = clk;
ri4.c = clk; ri3.c = clk; ri2.c = clk; ri1.c = clk;  ri0.c = clk;

ci9.c = clk; ci8.c = clk; ci7.c = clk; ci6.c = clk;  ci5.c = clk;
ci4.c = clk; ci3.c = clk; ci2.c = clk; ci1.c = clk;  ci0.c = clk;

match3.c = clk; match2.c = clk; match1.c = clk; match0.c = clk;

start.c = clk; cc.c = clk;

"#######################################################################
"                  equations  for ma(9:0)
"#######################################################################

ma9.d =
        (fct == [0,x,0]) & ma9.fb       "hold
      # (fct == [0,0,1]) & ri9.fb       "restart
      # (fct == [0,1,1]) & ci9.fb       "count
      # (fct == [1,0,x]) & ra9;         "load idle
                                        "load page  ma9 := 0

ma8.d =
        (fct == [0,x,0]) & ma8.fb       "hold
      # (fct == [0,0,1]) & ri8.fb       "restart
      # (fct == [0,1,1]) & ci8.fb       "count
      # (fct == [1,0,x]) & ra8          "load idle
      # (fct == [1,1,x]) & ca8;         "load page

ma7.d =
        (fct == [0,x,0]) & ma7.fb       "hold
      # (fct == [0,0,1]) & ri7.fb       "restart
      # (fct == [0,1,1]) & ci7.fb       "count
      # (fct == [1,0,x]) & ra7          "load idle
      # (fct == [1,1,x]) & ca7;         "load page

ma6.d =
        (fct == [0,x,0]) & ma6.fb       "hold
      # (fct == [0,0,1]) & ri6.fb       "restart
      # (fct == [0,1,1]) & ci6.fb       "count
      # (fct == [1,0,x]) & ra6          "load idle
      # (fct == [1,1,x]) & ca6;         "load page

ma5.d =
        (fct == [0,x,0]) & ma5.fb       "hold
      # (fct == [0,0,1]) & ri5.fb       "restart
      # (fct == [0,1,1]) & ci5.fb       "count
      # (fct == [1,0,x]) & ra5          "load idle
      # (fct == [1,1,x]) & ca5;          "load page

ma4.d =
        (fct == [0,x,0]) & ma4.fb        "hold
      # (fct == [0,0,1]) & ri4.fb        "restart
      # (fct == [0,1,1]) & ci4.fb        "count
      # (fct == [1,0,x]) & ra4           "load idle
      # (fct == [1,1,x]) & ca4;          "load page

ma3.d =
        (fct == [0,x,0]) & ma3.fb        "hold
      # (fct == [0,0,1]) & ri3.fb        "restart
      # (fct == [0,1,1]) & ci3.fb        "count
      # (fct == [1,0,x]) & ra3           "load idle
      # (fct == [1,1,x]) & ca3;          "load page

ma2.d =
        (fct == [0,x,0]) & ma2.fb        "hold
      # (fct == [0,0,1]) & ri2.fb        "restart
      # (fct == [0,1,1]) & ci2.fb        "count
      # (fct == [1,0,x]) & ra2           "load idle
      # (fct == [1,1,x]) & ca2;          "load page

ma1.d =
        (fct == [0,x,0]) & ma1.fb        "hold
      # (fct == [0,0,1]) & ri1.fb        "restart
      # (fct == [0,1,1]) & ci1.fb        "count
      # (fct == [1,0,x]) & ra1           "load idle
      # (fct == [1,1,x]) & ca1;          "load page

ma0.d =
        (fct == [0,x,0]) & ma0.fb        "hold
      # (fct == [0,0,1]) & ri0.fb        "restart
      # (fct == [0,1,1]) & ci0.fb        "count
      # (fct == [1,0,x]) & ra0           "load idle
      # (fct == [1,1,x]) & ca0;          "load page


"########################################################################
"             equations  for ci(9:0) NB: T-flip flops
"########################################################################


ci9.t =
        (fct == [0,0,1]) & !ma9.fb &  ci9.fb    "restart load ma
      # (fct == [0,0,1]) &  ma9.fb & !ci9.fb    "restart load ma
      # (fct == [1,0,x])           &  ci9.fb    "load idle ci9 := 0
      # (fct == [1,1,x])           &  ci9.fb;   "load page ci9 := 0

ci8.t =
        (fct == [0,0,1]) & !ma8.fb &  ci8.fb    "restart load ma
      # (fct == [0,0,1]) &  ma8.fb & !ci8.fb    "restart load ma
      # (fct == [1,0,x]) & !ca8    &  ci8.fb    "load idle load ca
      # (fct == [1,0,x]) &  ca8    & !ci8.fb    "load idle load ca
      # (fct == [1,1,x]) & !ca8    &  ci8.fb    "load page load ca
      # (fct == [1,1,x]) &  ca8    & !ci8.fb;   "load page load ca
```

```
ci7.t =
          (fct == [0,0,1]) & !ma7.fb &  ci7.fb    "restart load ma
        # (fct == [0,0,1]) &  ma7.fb & !ci7.fb    "restart load ma
        # ((fct == [0,1,1]) # (fct == [0,1,0]) & start.fb)"count
        & ci6.fb & ci5.fb & ci4.fb & ci3.fb & ci2.fb & ci1.fb & ci0.fb
        # (fct == [1,0,x]) & !ca7    &  ci7.fb    "load idle load ca
        # (fct == [1,0,x]) &  ca7    & !ci7.fb    "load idle load ca
        # (fct == [1,1,x]) & !ca7    &  ci7.fb    "load page load ca
        # (fct == [1,1,x]) &  ca7    & !ci7.fb;   "load page load ca

ci6.t =
          (fct == [0,0,1]) & !ma6.fb &  ci6.fb    "restart load ma
        # (fct == [0,0,1]) &  ma6.fb & !ci6.fb    "restart load ma
        # ((fct == [0,1,1]) # (fct == [0,1,0]) & start.fb)"count
        & ci5.fb & ci4.fb & ci3.fb & ci2.fb & ci1.fb & ci0.fb
        # (fct == [1,0,x]) & !ca6    &  ci6.fb    "load idle load ca
        # (fct == [1,0,x]) &  ca6    & !ci6.fb    "load idle load ca
        # (fct == [1,1,x]) & !ca6    &  ci6.fb    "load page load ca
        # (fct == [1,1,x]) &  ca6    & !ci6.fb;   "load page load ca

ci5.t =
          (fct == [0,0,1]) & !ma5.fb &  ci5.fb    "restart load ma
        # (fct == [0,0,1]) &  ma5.fb & !ci5.fb    "restart load ma
        # ((fct == [0,1,1]) # (fct == [0,1,0]) & start.fb)"count
        & ci4.fb & ci3.fb & ci2.fb & ci1.fb & ci0.fb
        # (fct == [1,0,x]) & !ca5    &  ci5.fb    "load idle load ca
        # (fct == [1,0,x]) &  ca5    & !ci5.fb    "load idle load ca
        # (fct == [1,1,x]) & !ca5    &  ci5.fb    "load page load ca
        # (fct == [1,1,x]) &  ca5    & !ci5.fb;   "load page load ca

ci4.t =
          (fct == [0,0,1]) & !ma4.fb &  ci4.fb    "restart load ma
        # (fct == [0,0,1]) &  ma4.fb & !ci4.fb    "restart load ma
        # ((fct == [0,1,1]) # (fct == [0,1,0]) & start.fb)"count
        & ci3.fb & ci2.fb & ci1.fb & ci0.fb
        # (fct == [1,0,x]) & !ca4    &  ci4.fb    "load idle load ca
        # (fct == [1,0,x]) &  ca4    & !ci4.fb    "load idle load ca
        # (fct == [1,1,x]) & !ca4    &  ci4.fb    "load page load ca
        # (fct == [1,1,x]) &  ca4    & !ci4.fb;   "load page load ca

ci3.t =
          (fct == [0,0,1]) & !ma3.fb &  ci3.fb    "restart load ma
        # (fct == [0,0,1]) &  ma3.fb & !ci3.fb    "restart load ma
        # ((fct == [0,1,1]) # (fct == [0,1,0]) & start.fb)"count
        & ci2.fb & ci1.fb & ci0.fb
        # (fct == [1,0,x]) & !ca3    &  ci3.fb    "load idle load ca
        # (fct == [1,0,x]) &  ca3    & !ci3.fb    "load idle load ca
        # (fct == [1,1,x]) & !ca3    &  ci3.fb    "load page load ca
        # (fct == [1,1,x]) &  ca3    & !ci3.fb;   "load page load ca

ci2.t =
          (fct == [0,0,1]) & !ma2.fb &  ci2.fb    "restart load ma
        # (fct == [0,0,1]) &  ma2.fb & !ci2.fb    "restart load ma
        # ((fct == [0,1,1]) # (fct == [0,1,0]) & start.fb)"count
        & ci1.fb & ci0.fb
```

```
      # (fct == [1,0,x]) & !ca2    &  ci2.fb    "load idle load ca
      # (fct == [1,0,x]) &  ca2    & !ci2.fb    "load idle load ca
      # (fct == [1,1,x]) & !ca2    &  ci2.fb    "load page load ca
      # (fct == [1,1,x]) &  ca2    & !ci2.fb;   "load page load ca

ci1.t =
        (fct == [0,0,1]) & !ma1.fb &  ci1.fb    "restart load ma
      # (fct == [0,0,1]) &  ma1.fb & !ci1.fb    "restart load ma
      # ((fct == [0,1,1]) # (fct == [0,1,0]) & start.fb)"count
      & ci0.fb
      # (fct == [1,0,x]) & !ca1    &  ci1.fb    "load idle load ca
      # (fct == [1,0,x]) &  ca1    & !ci1.fb    "load idle load ca
      # (fct == [1,1,x]) & !ca1    &  ci1.fb    "load page load ca
      # (fct == [1,1,x]) &  ca1    & !ci1.fb;   "load page load ca

ci0.t =
        (fct == [0,0,1]) & !ma0.fb &  ci0.fb    "restart load ma
      # (fct == [0,0,1]) &  ma0.fb & !ci0.fb    "restart load ma
      # ((fct == [0,1,1]) # (fct == [0,1,0]) & start.fb)"count
      # (fct == [1,0,x]) & !ca0    &  ci0.fb    "load idle load ca
      # (fct == [1,0,x]) &  ca0    & !ci0.fb    "load idle load ca
      # (fct == [1,1,x]) & !ca0    &  ci0.fb    "load page load ca
      # (fct == [1,1,x]) &  ca0    & !ci0.fb;   "load page load ca


cc.d  =
      !((fct == [0,0,1]) # (fct == [1,0,x]) # (fct == [1,1,x]))  "!load
      &!((fct == [0,1,1]) # (fct == [0,1,0]) & start.fb) & cc.fb "!count
      # ((fct == [0,0,1]) # (fct == [1,0,x]) # (fct == [1,1,x]))  " load
&ci7.fb & ci6.fb & ci5.fb & ci4.fb & ci3.fb & ci2.fb & ci1.fb & ci0.fb
      # ((fct == [0,1,1]) # (fct == [0,1,0]) & start.fb)         " count
&ci7.fb & ci6.fb & ci5.fb & ci4.fb & ci3.fb & ci2.fb & ci1.fb & !ci0.fb;



"#######################################################################
"                  equations  for ri(9:0)
"#######################################################################

ri9.d =
        (fct == [0,x,0]) & ri9.fb               "hold
      # (fct == [0,0,1]) & ri9.fb               "restart load ri
      # (fct == [0,1,1]) & ri9.fb               "count   hold ri
      # (fct == [1,0,x]) & ra9                  "load idle load ra
      # (fct == [1,1,x]) & ra9;                 "load page hold ra

ri8.d =
        (fct == [0,x,0]) & ri8.fb               "hold
      # (fct == [0,0,1]) & ri8.fb               "restart load ri
      # (fct == [0,1,1]) & ri8.fb               "count   hold ri
      # (fct == [1,0,x]) & ra8                  "load idle load ra
      # (fct == [1,1,x]) & ra8;                 "load page hold ra

ri7.d =
        (fct == [0,x,0]) & ri7.fb               "hold
      # (fct == [0,0,1]) & ri7.fb               "restart load ri
      # (fct == [0,1,1]) & ri7.fb               "count   hold ri
      # (fct == [1,0,x]) & ra7                  "load idle load ra
      # (fct == [1,1,x]) & ra7;                 "load page hold ra

ri6.d =
        (fct == [0,x,0]) & ri6.fb               "hold
      # (fct == [0,0,1]) & ri6.fb               "restart load ri
      # (fct == [0,1,1]) & ri6.fb               "count   hold ri
      # (fct == [1,0,x]) & ra6                  "load idle load ra
      # (fct == [1,1,x]) & ra6;                 "load page hold ra

ri5.d =
        (fct == [0,x,0]) & ri5.fb               "hold
      # (fct == [0,0,1]) & ri5.fb               "restart load ri
      # (fct == [0,1,1]) & ri5.fb               "count   hold ri
      # (fct == [1,0,x]) & ra5                  "load idle load ra
      # (fct == [1,1,x]) & ra5;                 "load page hold ra

ri4.d =
        (fct == [0,x,0]) & ri4.fb               "hold
      # (fct == [0,0,1]) & ri4.fb               "restart load ri
      # (fct == [0,1,1]) & ri4.fb               "count   hold ri
      # (fct == [1,0,x]) & ra4                  "load idle load ra
      # (fct == [1,1,x]) & ra4;                 "load page hold ra

ri3.d =
        (fct == [0,x,0]) & ri3.fb               "hold
      # (fct == [0,0,1]) & ri3.fb               "restart load ri
      # (fct == [0,1,1]) & ri3.fb               "count   hold ri
      # (fct == [1,0,x]) & ra3                  "load idle load ra
      # (fct == [1,1,x]) & ra3;                 "load page hold ra

ri2.d =
        (fct == [0,x,0]) & ri2.fb               "hold
      # (fct == [0,0,1]) & ri2.fb               "restart load ri
      # (fct == [0,1,1]) & ri2.fb               "count   hold ri
      # (fct == [1,0,x]) & ra2                  "load idle load ra
      # (fct == [1,1,x]) & ra2;                 "load page hold ra

ri1.d =
        (fct == [0,x,0]) & ri1.fb               "hold
      # (fct == [0,0,1]) & ri1.fb               "restart load ri
      # (fct == [0,1,1]) & ri1.fb               "count   hold ri
      # (fct == [1,0,x]) & ra1                  "load idle load ra
      # (fct == [1,1,x]) & ra1;                 "load page hold ra

ri0.d =
        (fct == [0,x,0]) & ri0.fb               "hold
      # (fct == [0,0,1]) & ri0.fb               "restart load ri
      # (fct == [0,1,1]) & ri0.fb               "count   hold ri
      # (fct == [1,0,x]) & ra0                  "load idle load ra
      # (fct == [1,1,x]) & ra0;                 "load page hold ra


"##########################################################################
#
"                         equations for match
```

```
  "#########################################################################
#

  match3.d =
    !ra9 & !ra8 & !ri9.fb & !ri8.fb & ale &!ra5 &!ri5.fb & !ra2 &!ri2.fb & al
e
  # !ra9 &  ra8 & !ri9.fb &  ri8.fb & ale &!ra5 &!ri5.fb & !ra2 &!ri2.fb & al
e
  #  ra9 & !ra8 &  ri9.fb & !ri8.fb & ale &!ra5 &!ri5.fb & !ra2 &!ri2.fb & al
e
  #  ra9 &  ra8 &  ri9.fb &  ri8.fb & ale &!ra5 &!ri5.fb & !ra2 &!ri2.fb & al
e
  # !ra9 & !ra8 & !ri9.fb & !ri8.fb & ale & ra5 & ri5.fb & !ra2 &!ri2.fb & al
e
  # !ra9 &  ra8 & !ri9.fb &  ri8.fb & ale & ra5 & ri5.fb & !ra2 &!ri2.fb & al
e
  #  ra9 & !ra8 &  ri9.fb & !ri8.fb & ale & ra5 & ri5.fb & !ra2 &!ri2.fb & al
e
  #  ra9 &  ra8 &  ri9.fb &  ri8.fb & ale & ra5 & ri5.fb & !ra2 &!ri2.fb & al
e
  # !ra9 & !ra8 & !ri9.fb & !ri8.fb & ale &!ra5 &!ri5.fb &  ra2 & ri2.fb & al
e
  # !ra9 &  ra8 & !ri9.fb &  ri8.fb & ale &!ra5 &!ri5.fb &  ra2 & ri2.fb & al
e
  #  ra9 & !ra8 &  ri9.fb & !ri8.fb & ale &!ra5 &!ri5.fb &  ra2 & ri2.fb & al
e
  #  ra9 &  ra8 &  ri9.fb &  ri8.fb & ale &!ra5 &!ri5.fb &  ra2 & ri2.fb & al
e
  # !ra9 & !ra8 & !ri9.fb & !ri8.fb & ale & ra5 & ri5.fb &  ra2 & ri2.fb & al
e
  # !ra9 &  ra8 & !ri9.fb &  ri8.fb & ale & ra5 & ri5.fb &  ra2 & ri2.fb & al
e
  #  ra9 & !ra8 &  ri9.fb & !ri8.fb & ale & ra5 & ri5.fb &  ra2 & ri2.fb & al
e
  #  ra9 &  ra8 &  ri9.fb &  ri8.fb & ale & ra5 & ri5.fb &  ra2 & ri2.fb & al
e;


  match2.d =
    !ra7 & !ra6 & !ri7.fb & !ri6.fb & ale
  # !ra7 &  ra6 & !ri7.fb &  ri6.fb & ale
  #  ra7 & !ra6 &  ri7.fb & !ri6.fb & ale
  #  ra7 &  ra6 &  ri7.fb &  ri6.fb & ale;

  match1.d =
     !ra4 & !ra3 & !ri4.fb & !ri3.fb & ale
   # !ra4 &  ra3 & !ri4.fb &  ri3.fb & ale
   #  ra4 & !ra3 &  ri4.fb & !ri3.fb & ale
   #  ra4 &  ra3 &  ri4.fb &  ri3.fb & ale;


  match0.d =
     !ra1 & !ra0 & !ri1.fb & !ri0.fb & ale
   # !ra1 &  ra0 & !ri1.fb &  ri0.fb & ale
   #  ra1 & !ra0 &  ri1.fb & !ri0.fb & ale
   #  ra1 &  ra0 &  ri1.fb &  ri0.fb & ale;

  start.d = ale;

  declarations;

  ra = [ra9,ra8,ra7,ra6,ra5,ra4,ra3,ra2,ra1,ra0];
```

```
ca =     [ca8,ca7,ca6,ca5,ca4,ca3,ca2,ca1,ca0];
ma = [ma9,ma8,ma7,ma6,ma5,ma4,ma3,ma2,ma1,ma0];
ri = [ri9,ri8,ri7,ri6,ri5,ri4,ri3,ri2,ri1,ri0];
ci = [ci9,ci8,ci7,ci6,ci5,ci4,ci3,ci2,ci1,ci0];

test_vectors
([clk,ale,fct1,fct0,ra,ca]
-> [ma, ri, ci]);
"  1 1 0    ra     ca           ma          ri      ci
[c,1,0,x, ^h000, ^h0ff] -> [ ^h000,       ^h000, ^h0ff       ];"load
[c,0,1,1, ^h111, ^h022] -> [ ^h0ff,       ^h000, ^h000       ];"mux
```

```
[c,0,1,1, ^h111, ^h022] -> [ ^h000,      ^h000, ^h001      ];"mux
[c,0,x,0, ^h111, ^h022] -> [ ^h000,      ^h000, ^h001      ];"hold
"  1 1 0     ra      ca          ma          ri      ci
[c,1,0,x, ^h000, ^h1ff] -> [ ^h000,      ^h000, ^h1ff      ];"load
[c,0,1,1, ^h111, ^h022] -> [ ^h1ff,      ^h000, ^h100      ];"mux
[c,0,1,1, ^h111, ^h022] -> [ ^h100,      ^h000, ^h101      ];"mux
[c,0,x,0, ^h111, ^h022] -> [ ^h100,      ^h000, ^h101      ];"hold
"  1 1 0     ra      ca          ma          ri      ci
[c,1,0,x, ^h111, ^h0ee] -> [ ^h111,      ^h111, ^h0ee      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h0ee,      ^h111, ^h0ee + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h0ee + 1,  ^h111, ^h0ee + 2  ];"mux
[c,0,x,0, ^haaa, ^haaa] -> [ ^h0ee + 1,  ^h111, ^h0ee + 2  ];"hold
"  1 1 0     ra      ca          ma          ri      ci
[c,1,0,x, ^h222, ^h1dd] -> [ ^h222,      ^h222, ^h1dd      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h1dd,      ^h222, ^h1dd + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h1dd + 1,  ^h222, ^h1dd + 2  ];"mux
[c,0,x,0, ^haaa, ^haaa] -> [ ^h1dd + 1,  ^h222, ^h1dd + 2  ];"hold
"  1 1 0     ra      ca          ma          ri      ci
[c,1,0,x, ^h333, ^h0cc] -> [ ^h333,      ^h333, ^h0cc      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h0cc,      ^h333, ^h0cc + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h0cc + 1,  ^h333, ^h0cc + 2  ];"mux
[c,0,x,0, ^haaa, ^haaa] -> [ ^h0cc + 1,  ^h333, ^h0cc + 2  ];"hold
"  1 1 0     ra      ca          ma          ri      ci
[c,1,0,x, ^h444, ^h1bb] -> [ ^h444,      ^h444, ^h1bb      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h1bb,      ^h444, ^h1bb + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h1bb + 1,  ^h444, ^h1bb + 2  ];"mux
[c,0,x,0, ^haaa, ^haaa] -> [ ^h1bb + 1,  ^h444, ^h1bb + 2  ];"hold
"  1 1 0     ra      ca          ma          ri      ci
[c,1,0,x, ^h555, ^h0aa] -> [ ^h555,      ^h555, ^h0aa      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h0aa,      ^h555, ^h0aa + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h0aa + 1,  ^h555, ^h0aa + 2  ];"mux
[c,0,x,0, ^haaa, ^haaa] -> [ ^h0aa + 1,  ^h555, ^h0aa + 2  ];"hold
"  1 1 0     ra      ca          ma          ri      ci
[c,1,0,x, ^h666, ^h199] -> [ ^h666,      ^h666, ^h199      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h199,      ^h666, ^h199 + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h199 + 1,  ^h666, ^h199 + 2  ];"mux
[c,0,x,0, ^haaa, ^haaa] -> [ ^h199 + 1,  ^h666, ^h199 + 2  ];"hold
"  1 1 0     ra      ca          ma          ri      ci
[c,1,0,x, ^h777, ^h088] -> [ ^h777,      ^h777, ^h088      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h088,      ^h777, ^h088 + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h088 + 1,  ^h777, ^h088 + 2  ];"mux
[c,0,x,0, ^haaa, ^haaa] -> [ ^h088 + 1,  ^h777, ^h088 + 2  ];"hold
"  1 1 0     ra      ca          ma          ri      ci
[c,1,0,x, ^h888, ^h177] -> [ ^h888,      ^h888, ^h177      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h177,      ^h888, ^h177 + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h177 + 1,  ^h888, ^h177 + 2  ];"mux
[c,0,x,0, ^haaa, ^haaa] -> [ ^h177 + 1,  ^h888, ^h177 + 2  ];"hold
"  1 1 0     ra      ca          ma          ri      ci
[c,1,0,x, ^h999, ^h066] -> [ ^h999,      ^h999, ^h066      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h066,      ^h999, ^h066 + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h066 + 1,  ^h999, ^h066 + 2  ];"mux
[c,0,x,0, ^haaa, ^haaa] -> [ ^h066 + 1,  ^h999, ^h066 + 2  ];"hold
"  1 1 0     ra      ca          ma          ri      ci
[c,1,0,x, ^haaa, ^h155] -> [ ^haaa,      ^haaa, ^h155      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h155,      ^haaa, ^h155 + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h155 + 1,  ^haaa, ^h155 + 2  ];"mux
```

```
[c,0,x,0, ^haaa, ^haaa] -> [ ^h155 + 1,  ^haaa, ^h155 + 2  ];"hold
"  1 1 0    ra     ca          ma          ri     ci
[c,1,0,x, ^hbbb, ^h044] -> [ ^hbbb,      ^hbbb, ^h044      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h044,      ^hbbb, ^h044 + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h044 + 1,  ^hbbb, ^h044 + 2  ];"mux
[c,0,x,0, ^haaa, ^haaa] -> [ ^h044 + 1,  ^hbbb, ^h044 + 2  ];"hold
"  1 1 0    ra     ca          ma          ri     ci
[c,1,0,x, ^hccc, ^h133] -> [ ^hccc,      ^hccc, ^h133      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h133,      ^hccc, ^h133 + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h133 + 1,  ^hccc, ^h133 + 2  ];"mux
[c,0,x,0, ^haaa, ^haaa] -> [ ^h133 + 1,  ^hccc, ^h133 + 2  ];"hold
"  1 1 0    ra     ca          ma          ri     ci
[c,1,0,x, ^hddd, ^h022] -> [ ^hddd,      ^hddd, ^h022      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h022,      ^hddd, ^h022 + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h022 + 1,  ^hddd, ^h022 + 2  ];"mux
[c,0,x,0, ^haaa, ^haaa] -> [ ^h022 + 1,  ^hddd, ^h022 + 2  ];"hold
"  1 1 0    ra     ca          ma          ri     ci
[c,1,0,x, ^heee, ^h111] -> [ ^heee,      ^heee, ^h111      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h111,      ^heee, ^h111 + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h111 + 1,  ^heee, ^h111 + 2  ];"mux
[c,0,x,0, ^haaa, ^haaa] -> [ ^h111 + 1,  ^heee, ^h111 + 2  ];"hold
"  1 1 0    ra     ca          ma          ri     ci
[c,1,0,x, ^hfff, ^h000] -> [ ^hfff,      ^hfff, ^h000      ];"load
[c,0,1,1, ^haaa, ^haaa] -> [ ^h000,      ^hfff, ^h000 + 1  ];"mux
[c,0,1,1, ^haaa, ^haaa] -> [ ^h000 + 1,  ^hfff, ^h000 + 2  ];"mux
[c,0,x,0, ^haaa, ^haaa] -> [ ^h000 + 1,  ^hfff, ^h000 + 2  ];"hold

"  1 1 0    ra     ca          ma          ri     ci
[c,1,1,x, ^h000, ^h0ff] -> [ ^h0ff,      ^h000, ^h0ff      ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h0ff,      ^h000, ^h000      ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^h000,      ^h000, ^h0ff      ];"restart
"  1 1 0    ra     ca          ma          ri     ci
[c,1,1,x, ^h000, ^h1ff] -> [ ^h1ff,      ^h000, ^h1ff      ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h1ff,      ^h000, ^h100      ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^h000,      ^h000, ^h1ff      ];"restart
"  1 1 0    ra     ca          ma          ri     ci
[c,1,1,x, ^h111, ^h1ee] -> [ ^h1ee,      ^h111, ^h1ee      ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h1ee,      ^h111, ^h1ee + 1  ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^h111,      ^h111, ^h1ee      ];"restart
"  1 1 0    ra     ca          ma          ri     ci
[c,1,1,x, ^h222, ^h0dd] -> [ ^h0dd,      ^h222, ^h0dd      ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h0dd,      ^h222, ^h0dd + 1  ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^h222,      ^h222, ^h0dd      ];"restart
"  1 1 0    ra     ca          ma          ri     ci
[c,1,1,x, ^h333, ^h1cc] -> [ ^h1cc,      ^h333, ^h1cc      ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h1cc,      ^h333, ^h1cc + 1  ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^h333,      ^h333, ^h1cc      ];"restart
"  1 1 0    ra     ca          ma          ri     ci
[c,1,1,x, ^h444, ^h0bb] -> [ ^h0bb,      ^h444, ^h0bb      ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h0bb,      ^h444, ^h0bb + 1  ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^h444,      ^h444, ^h0bb      ];"restart
"  1 1 0    ra     ca          ma          ri     ci
[c,1,1,x, ^h555, ^h1aa] -> [ ^h1aa,      ^h555, ^h1aa      ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h1aa,      ^h555, ^h1aa + 1  ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^h555,      ^h555, ^h1aa      ];"restart
"  1 1 0    ra     ca          ma          ri     ci
```

```
[c,1,1,x, ^h666, ^h099] -> [ ^h099,      ^h666, ^h099       ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h099,      ^h666, ^h099 + 1   ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^h666,      ^h666, ^h099       ];"restart
"  1 1 0   ra     ca           ma          ri     ci
[c,1,1,x, ^h777, ^h188] -> [ ^h188,      ^h777, ^h188       ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h188,      ^h777, ^h188 + 1   ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^h777,      ^h777, ^h188       ];"restart
"  1 1 0   ra     ca           ma          ri     ci
[c,1,1,x, ^h888, ^h077] -> [ ^h077,      ^h888, ^h077       ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h077,      ^h888, ^h077 + 1   ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^h888,      ^h888, ^h077       ];"restart
"  1 1 0   ra     ca           ma          ri     ci
[c,1,1,x, ^h999, ^h166] -> [ ^h166,      ^h999, ^h166       ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h166,      ^h999, ^h166 + 1   ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^h999,      ^h999, ^h166       ];"restart
"  1 1 0   ra     ca           ma          ri     ci
[c,1,1,x, ^haaa, ^h055] -> [ ^h055,      ^haaa, ^h055       ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h055,      ^haaa, ^h055 + 1   ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^haaa,      ^haaa, ^h055       ];"restart
"  1 1 0   ra     ca           ma          ri     ci
[c,1,1,x, ^hbbb, ^h144] -> [ ^h144,      ^hbbb, ^h144       ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h144,      ^hbbb, ^h144 + 1   ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^hbbb,      ^hbbb, ^h144       ];"restart
"  1 1 0   ra     ca           ma          ri     ci
[c,1,1,x, ^hccc, ^h033] -> [ ^h033,      ^hccc, ^h033       ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h033,      ^hccc, ^h033 + 1   ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^hccc,      ^hccc, ^h033       ];"restart
"  1 1 0   ra     ca           ma          ri     ci
[c,1,1,x, ^hddd, ^h122] -> [ ^h122,      ^hddd, ^h122       ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h122,      ^hddd, ^h122 + 1   ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^hddd,      ^hddd, ^h122       ];"restart
"  1 1 0   ra     ca           ma          ri     ci
[c,1,1,x, ^heee, ^h011] -> [ ^h011,      ^heee, ^h011       ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h011,      ^heee, ^h011 + 1   ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^heee,      ^heee, ^h011       ];"restart
"  1 1 0   ra     ca           ma          ri     ci
[c,1,1,x, ^hfff, ^h100] -> [ ^h100,      ^hfff, ^h100       ];"load pa
[c,0,1,0, ^h111, ^h022] -> [ ^h100,      ^hfff, ^h100 + 1   ];"hold
[c,0,0,1, ^h111, ^h022] -> [ ^hfff,      ^hfff, ^h100       ];"restart

test_vectors([clk,ale,ra9,ra8,ra5,ra2,ra7,ra6,ra4,ra3,ra1,ra0] ->
[match3,match2,match1,match0]);
[c,1,0,0,0,0, 0,0, 0,0, 0,0] -> [x,x,x,x];
[c,1,0,0,0,0, 0,0, 0,0, 0,0] -> [1,1,1,1];
[c,1,0,1,0,1, 0,1, 0,1, 0,1] -> [0,0,0,0];
[c,1,0,1,0,1, 0,1, 0,1, 0,1] -> [1,1,1,1];
[c,1,1,1,1,1, 1,1, 1,1, 1,1] -> [0,0,0,0];
[c,1,1,1,1,1, 1,1, 1,1, 1,1] -> [1,1,1,1];
[c,1,1,0,1,0, 1,0, 1,0, 1,0] -> [0,0,0,0];
[c,1,1,0,1,0, 1,0, 1,0, 1,0] -> [1,1,1,1];
[c,1,0,0,0,0, 0,0, 0,0, 0,0] -> [0,0,0,0];
[c,1,1,1,1,1, 1,1, 1,1, 1,1] -> [0,0,0,0];
[c,1,0,0,0,0, 0,0, 0,0, 0,0] -> [0,0,0,0];
[c,1,1,0,1,0, 1,0, 1,0, 1,0] -> [0,0,0,0];
[c,1,1,1,1,1, 1,1, 1,1, 1,1] -> [0,0,0,0];
[c,1,0,1,0,1, 0,1, 0,1, 0,1] -> [0,0,0,0];
```

```
[c,1,1,0,1,0, 1,0, 1,0, 1,0] -> [0,0,0,0];
[c,1,0,1,0,1, 0,1, 0,1, 0,1] -> [0,0,0,0];
[c,1,0,0,0,0, 0,0, 0,0, 0,0] -> [0,0,0,0];

[c,1,0,0,0,0, 0,0, 0,0, 0,0] -> [1,1,1,1];
[c,1,0,0,0,0, 0,0, 0,0, 0,0] -> [1,1,1,1];
[c,1,0,0,0,1, 0,1, 0,1, 0,1] -> [0,0,0,0];
[c,1,0,0,0,1, 0,1, 0,1, 0,1] -> [1,1,1,1];
[c,1,0,0,1,1, 1,1, 1,1, 1,1] -> [0,0,0,0];
[c,1,0,0,1,1, 1,1, 1,1, 1,1] -> [1,1,1,1];
[c,1,0,0,1,0, 1,0, 1,0, 1,0] -> [0,0,0,0];
[c,1,0,0,1,0, 1,0, 1,0, 1,0] -> [1,1,1,1];
[c,1,0,1,1,0, 0,0, 0,0, 0,0] -> [0,0,0,0];
[c,1,0,1,1,0, 1,1, 1,1, 1,1] -> [1,0,0,0];
[c,1,0,1,1,1, 0,0, 0,0, 0,0] -> [0,0,0,0];
[c,1,0,1,1,1, 1,0, 1,0, 1,0] -> [1,0,0,0];
[c,1,0,1,0,1, 1,1, 1,1, 1,1] -> [0,0,0,0];
[c,1,0,1,0,1, 0,1, 0,1, 0,1] -> [1,0,0,0];
[c,1,0,1,0,0, 1,0, 1,0, 1,0] -> [0,0,0,0];
[c,1,0,1,0,0, 0,1, 0,1, 0,1] -> [1,0,0,0];


[c,1,1,1,0,0, 0,0, 0,0, 0,0] -> [0,0,0,0];
[c,1,1,1,0,0, 0,0, 0,0, 0,0] -> [1,1,1,1];
[c,1,1,1,0,1, 0,1, 0,1, 0,1] -> [0,0,0,0];
[c,1,1,1,0,1, 0,1, 0,1, 0,1] -> [1,1,1,1];
[c,1,1,1,1,1, 1,1, 1,1, 1,1] -> [0,0,0,0];
[c,1,1,1,1,1, 1,1, 1,1, 1,1] -> [1,1,1,1];
[c,1,1,1,1,0, 1,0, 1,0, 1,0] -> [0,0,0,0];
[c,1,1,1,1,0, 1,0, 1,0, 1,0] -> [1,1,1,1];
[c,1,1,0,1,0, 0,0, 0,0, 0,0] -> [0,0,0,0];
[c,1,1,0,1,0, 1,1, 1,1, 1,1] -> [1,0,0,0];
[c,1,1,0,1,1, 0,0, 0,0, 0,0] -> [0,0,0,0];
[c,1,1,0,1,1, 1,0, 1,0, 1,0] -> [1,0,0,0];
[c,1,1,0,0,1, 1,1, 1,1, 1,1] -> [0,0,0,0];
[c,1,1,0,0,1, 0,1, 0,1, 0,1] -> [1,0,0,0];
[c,1,1,0,0,0, 1,0, 1,0, 1,0] -> [0,0,0,0];
[c,1,1,0,0,0, 0,1, 0,1, 0,1] -> [1,0,0,0];

end drama;
```

```
----------------------------------------------------

User pre-assignments:
    ma0                     PIN 4
    ma1                     PIN 6
    ma2                     PIN 8
    ma3                     PIN 19
    ma4                     PIN 17
    ma5                     PIN 15
    ma6                     PIN 24
    ma7                     PIN 30
    ma8                     PIN 43
    ma9                     PIN 37
    match0                  PIN 9
    match1                  PIN 14
    match2                  PIN 28
    match3                  PIN 39
    ri0                     NODE 46
    ri1                     NODE 48
    ri2                     NODE 52
    ri3                     NODE 54
    ri4                     NODE 56
    ri5                     NODE 60
    ri6                     NODE 61
    ri7                     NODE 68
    ri8                     NODE 69
    ri9                     NODE 76
    ci0                     NODE 45
    ci1                     NODE 47
    ci2                     NODE 50
    ci3                     NODE 53
    ci4                     NODE 55
    ci5                     NODE 58
    ci6                     NODE 63
    ci7                     NODE 67
    ci8                     NODE 71
    ci9                     NODE 75

---------------------------------------------------
Results of fitting drama.tt2 into MACH210.
---------------------------------------------------

---------- Dedicated Inputs --------
PIN:10  --> ra5
PIN:11  --> ale
PIN:32  --> fct1
PIN:33  --> fct0

---------- Clock Inputs ------------
PIN:13  --> clk
PIN:35  --> ra2


------------------------------------
       Resources of block A
------------------------------------
```

```
Inputs routed to block:  ra2 ra1 ra0 ca2 ca1 ca0 ale fct1 fct0 ma2.FB
ri2.FB ci2.FB ma1.FB ri1.FB ci1.FB ma0.FB ri0.FB ci0.FB start.FB

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  A0   (PIN :2   ) -->
      PIN INPUT             : ca8
      PRODUCT TERMS         : DOWN
  A1   (NODE:45 ) -->
      INTERNAL FEEDBACK : ci0.T (pts=6)
      PRODUCT TERMS         : LOCAL
  A2   (PIN :3   ) -->
      PIN INPUT             : ra8
      PRODUCT TERMS         : DOWN
  A3   (NODE:46 ) -->
      INTERNAL FEEDBACK : ri0.D (pts=2)
      PRODUCT TERMS         : DOWN
  A4   (PIN :4   ) -->
      PIN OUTPUT            : ma0.D (pts=5)
      PRODUCT TERMS         : DOWN
  A5   (NODE:47 ) -->
      INTERNAL FEEDBACK : ci1.T (pts=6)
      PRODUCT TERMS         : LOCAL
  A6   (PIN :5   ) -->
      PIN INPUT             : ra9
      PRODUCT TERMS         : UP2
  A7   (NODE:48 ) -->
      INTERNAL FEEDBACK : ri1.D (pts=2)
      PRODUCT TERMS         : LOCAL
  A8   (PIN :6   ) -->
      PIN OUTPUT            : ma1.D (pts=5)
      PRODUCT TERMS         : LOCAL
  A9   (NODE:49 ) -->
      PRODUCT TERMS         : UP1
  A10  (PIN :7   ) -->
      PIN INPUT             : ca6
      PRODUCT TERMS         : DOWN
  A11  (NODE:50 ) -->
      INTERNAL FEEDBACK : ci2.T (pts=6)
      PRODUCT TERMS         : DOWN
  A12  (PIN :8   ) -->
      PIN OUTPUT            : ma2.D (pts=5)
      PRODUCT TERMS         : LOCAL
  A13  (NODE:51 ) -->
      PRODUCT TERMS         : UP2
  A14  (PIN :9   ) -->
      PIN OUTPUT            : match0.D (pts=4)
      PRODUCT TERMS         : LOCAL
  A15  (NODE:52 ) -->
```

```
      INTERNAL FEEDBACK : ri2.D (pts=2)
      PRODUCT TERMS     : LOCAL

-----------------------------------
       Block A resource summary:

                 USED     FREE     UTILIZATION
  product terms  43       21        67.2%
  macro cells    16        0       100.0%
  pins            8        0       100.0%
-----------------------------------



-----------------------------------
       Resources of block B
-----------------------------------
Inputs routed to block:  ra5 ra4 ra3 ca5 ca4 ca3 ale fct1 fct0 ma5.FB
ri5.FB ci5.FB ma4.FB ri4.FB ci4.FB ma3.FB ri3.FB ci3.FB ci2.FB ci1.FB
ci0.FB start.FB

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  B0   (PIN :21 ) -->
      PIN INPUT         : ca7
      PRODUCT TERMS     : DOWN
  B1   (NODE:53 ) -->
      INTERNAL FEEDBACK : ci3.T (pts=6)
      PRODUCT TERMS     : LOCAL
  B2   (PIN :20 ) -->
      PIN INPUT         : ra6
      PRODUCT TERMS     : DOWN
  B3   (NODE:54 ) -->
      INTERNAL FEEDBACK : ri3.D (pts=2)
      PRODUCT TERMS     : DOWN
  B4   (PIN :19 ) -->
      PIN OUTPUT        : ma3.D (pts=5)
      PRODUCT TERMS     : DOWN
  B5   (NODE:55 ) -->
      INTERNAL FEEDBACK : ci4.T (pts=6)
      PRODUCT TERMS     : LOCAL
  B6   (PIN :18 ) -->
      PIN INPUT         : ra7
      PRODUCT TERMS     : UP2
  B7   (NODE:56 ) -->
      INTERNAL FEEDBACK : ri4.D (pts=2)
      PRODUCT TERMS     : LOCAL
  B8   (PIN :17 ) -->
      PIN OUTPUT        : ma4.D (pts=5)
      PRODUCT TERMS     : LOCAL
  B9   (NODE:57 ) -->
      PRODUCT TERMS     : UP1
  B10  (PIN :16 ) -->
      PIN INPUT         : ca3
      PRODUCT TERMS     : DOWN
  B11  (NODE:58 ) -->
      INTERNAL FEEDBACK : ci5.T (pts=6)
      PRODUCT TERMS     : DOWN
  B12  (PIN :15 ) -->
      PIN OUTPUT        : ma5.D (pts=5)
      PRODUCT TERMS     : LOCAL
  B13  (NODE:59 ) -->
      PRODUCT TERMS     : UP2
  B14  (PIN :14 ) -->
      PIN OUTPUT        : match1.D (pts=4)
      PRODUCT TERMS     : LOCAL
  B15  (NODE:60 ) -->
      INTERNAL FEEDBACK : ri5.D (pts=2)
      PRODUCT TERMS     : LOCAL

------------------------------------
       Block B resource summary:

              USED    FREE    UTILIZATION
  product terms 43      21       67.2%
  macro cells   16       0      100.0%
  pins           8       0      100.0%
------------------------------------



------------------------------------
       Resources of block C
------------------------------------
Inputs routed to block:  ra7 ra6 ca7 ca6 ale fct1 fct0 ma7.FB ri7.FB ci7.FB
ma6.FB ri6.FB ci6.FB ci5.FB ci4.FB ci3.FB ci2.FB ci1.FB ci0.FB start.FB
cc.FB

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  C0    (PIN :24 ) -->
      PIN OUTPUT        : ma6.D (pts=5)
      PRODUCT TERMS     : LOCAL
  C1   (NODE:61 ) -->
      INTERNAL FEEDBACK : ri6.D (pts=2)
      PRODUCT TERMS     : UP1
  C2   (PIN :25 ) -->
      PIN INPUT         : ca4
      PRODUCT TERMS      : UP1
  C3   (NODE:62 ) -->
      PRODUCT TERMS      : NOT USED
  C4   (PIN :26 ) -->
      PIN INPUT          : ca5
      PRODUCT TERMS      : DOWN
  C5   (NODE:63 ) -->
      INTERNAL FEEDBACK : ci6.T (pts=6)
      PRODUCT TERMS      : LOCAL
  C6   (PIN :27 ) -->
      PIN INPUT          : ra3
      PRODUCT TERMS      : DOWN
  C7   (NODE:64 ) -->
      INTERNAL FEEDBACK : cc.D (pts=6)
      PRODUCT TERMS      : LOCAL
  C8   (PIN :28 ) -->
      PIN OUTPUT         : match2.D (pts=4)
      PRODUCT TERMS      : LOCAL
  C9   (NODE:65 ) -->
      PRODUCT TERMS      : NOT USED
  C10  (PIN :29 ) -->
      PIN INPUT          : ca0
      PRODUCT TERMS      : NOT USED
  C11  (NODE:66 ) -->
      PRODUCT TERMS      : DOWN
  C12  (PIN :30 ) -->
      PIN OUTPUT         : ma7.D (pts=5)
      PRODUCT TERMS      : LOCAL
  C13  (NODE:67 ) -->
      INTERNAL FEEDBACK : ci7.T (pts=6)
      PRODUCT TERMS      : LOCAL
  C14  (PIN :31 ) -->
      PIN INPUT          : ra4
      PRODUCT TERMS      : UP1
  C15  (NODE:68 ) -->
      INTERNAL FEEDBACK : ri7.D (pts=2)
      PRODUCT TERMS      : LOCAL

------------------------------------
       Block C resource summary:

                   USED     FREE    UTILIZATION
  product terms  36       28       56.3%
  macro cells    13        3       81.3%
  pins            8        0       100.0%
------------------------------------



------------------------------------
       Resources of block D
------------------------------------
Inputs routed to block:  ra9 ra8 ra5 ra2 ca8 ale fct1 fct0 ma9.FB ri9.FB
ci9.FB ma8.FB ri8.FB ci8.FB ri5.FB ri2.FB

Controls on block:
   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  D0   (PIN :43 ) -->
      PIN OUTPUT           : ma8.D (pts=5)
      PRODUCT TERMS        : LOCAL
  D1   (NODE:69 ) -->
      INTERNAL FEEDBACK : ri8.D (pts=2)
      PRODUCT TERMS        : UP1
  D2   (PIN :42 ) -->
      PIN OUTPUT           : start.D (pts=1)
      PRODUCT TERMS        : UP1
  D3   (NODE:70 ) -->
      PRODUCT TERMS        : UP1
  D4   (PIN :41 ) -->
      PIN INPUT            : ca1
      PRODUCT TERMS        : NOT USED
  D5   (NODE:71 ) -->
      INTERNAL FEEDBACK : ci8.T (pts=4)
      PRODUCT TERMS        : LOCAL
  D6   (PIN :40 ) -->
      PIN INPUT            : ca2
      PRODUCT TERMS        : NOT USED
  D7   (NODE:72 ) -->
      PRODUCT TERMS        : DOWN
  D8   (PIN :39 ) -->
      PIN OUTPUT           : match3.D (pts=16)
      PRODUCT TERMS        : LOCAL
  D9   (NODE:73 ) -->
      PRODUCT TERMS        : UP1
  D10  (PIN :38 ) -->
      PIN INPUT            : ra0
      PRODUCT TERMS        : UP2
  D11  (NODE:74 ) -->
      PRODUCT TERMS        : NOT USED
  D12  (PIN :37 ) -->
      PIN OUTPUT           : ma9.D (pts=4)
      PRODUCT TERMS        : LOCAL
  D13  (NODE:75 ) -->
      INTERNAL FEEDBACK : ci9.T (pts=3)
      PRODUCT TERMS        : LOCAL
  D14  (PIN :36 ) -->
      PIN INPUT            : ra1
      PRODUCT TERMS        : NOT USED
  D15  (NODE:76 ) -->
      INTERNAL FEEDBACK : ri9.D (pts=2)
      PRODUCT TERMS        : LOCAL

------------------------------------
       Block D resource summary:
```

```
                USED    FREE    UTILIZATION
  product terms  37      27      57.8%
  macro cells    12       4      75.0%
  pins            8       0      100.0%
------------------------------------



------------------------------------
      CHIP resource summary:

                USED    FREE    UTILIZATION
  product terms 159      97      62.1%
  macro cells    57       7      89.1%
  pins           38       0      100.0%

------------------------------------

----------------------------------------------------
Routing results.
----------------------------------------------------

---------------------
 Routes for block A
---------------------
+----------------------------------------------+
|input -->           path               : signal |
+----------------------------------------------+
|   0  | --> pin route of C10           : ca0
|   1  | --> internal route of D2       : start.FB
|   2  | -->
|   3  | -->
|   4  | --> internal route of A3       : ri0.FB
|   5  | --> internal route of A11      : ci2.FB
|   6  | --> pin route of INPUT3        : fct1
|   7  | --> pin route of INPUT4        : fct0
|   8  | --> internal route of A7       : ri1.FB
|   9  | --> internal route of A15      : ri2.FB
|  10  | -->
|  11  | --> internal route of A4       : ma0.FB
|  12  | --> pin route of D10           : ra0
|  13  | --> pin route of INPUT5        : ra2
|  14  | --> internal route of A1       : ci0.FB
|  15  | --> internal route of A8       : ma1.FB
|  16  | --> pin route of INPUT1        : ale
|  17  | --> pin route of D14           : ra1
|  18  | --> internal route of A5       : ci1.FB
|  19  | --> internal route of A12      : ma2.FB
|  20  | --> pin route of D4            : ca1
|  21  | --> pin route of D6            : ca2

19 of 22 routes used (utilization = 86.4%)

---------------------
 Routes for block B
---------------------
```

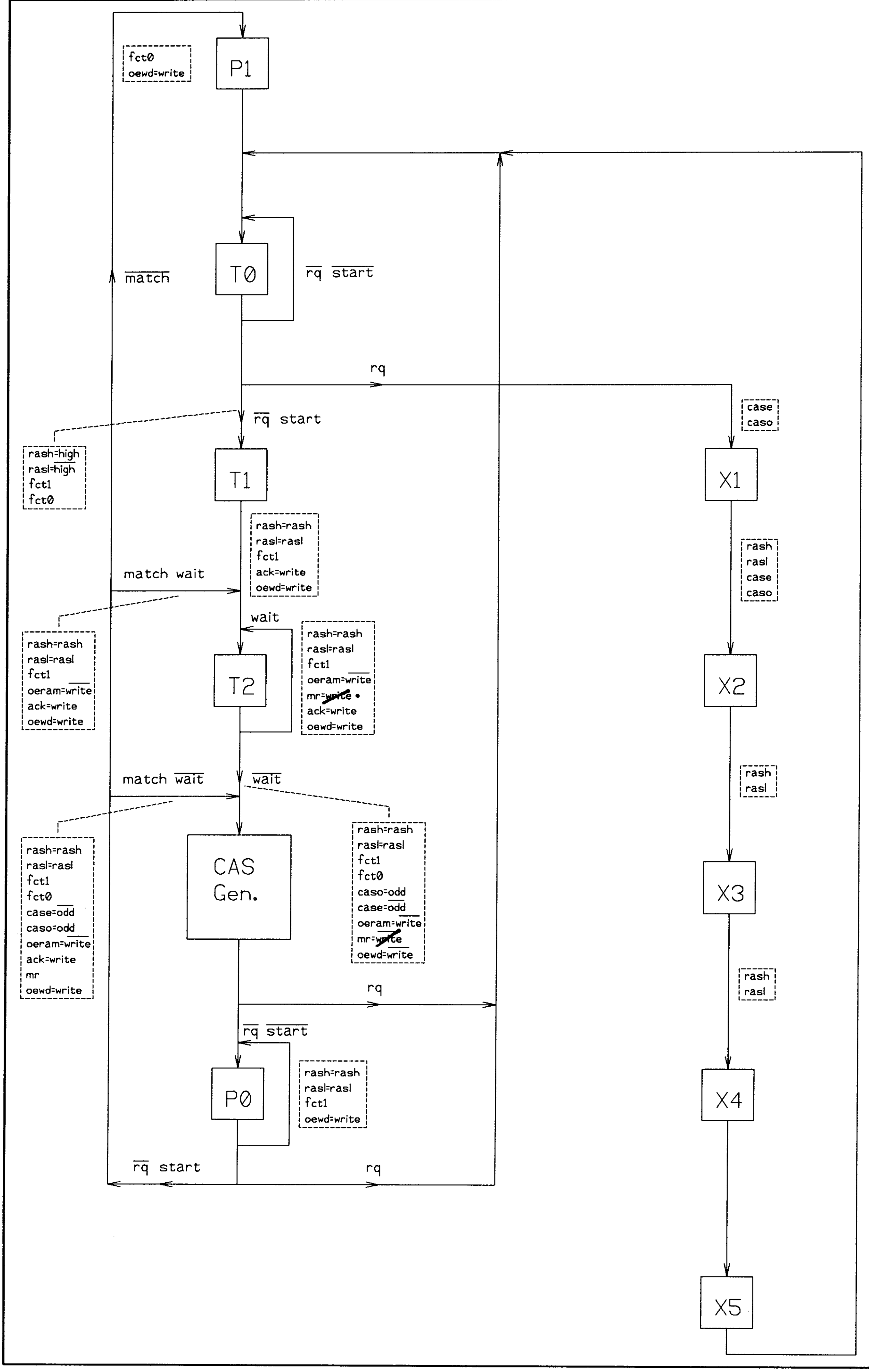
```
+-----------------------------------------------+
|input -->           path             : signal |
+-----------------------------------------------+
|   0  | --> internal route of B7     : ri4.FB
|   1  | --> internal route of D2     : start.FB
|   2  | --> internal route of A1     : ci0.FB
|   3  | --> pin route of INPUT3      : fct1
|   4  | --> pin route of INPUT4      : fct0
|   5  | --> internal route of A11    : ci2.FB
|   6  | --> internal route of A5     : ci1.FB
|   7  | --> internal route of B3     : ri3.FB
|   8  | --> internal route of B11    : ci5.FB
|   9  | --> internal route of B5     : ci4.FB
|  10  | --> pin route of INPUT0      : ra5
|  11  | --> pin route of B10         : ca3
|  12  | --> internal route of B4     : ma3.FB
|  13  | --> pin route of C2          : ca4
|  14  | --> pin route of C4          : ca5
|  15  | --> pin route of C6          : ra3
|  16  | --> pin route of INPUT1      : ale
|  17  | --> internal route of B15    : ri5.FB
|  18  | --> internal route of B12    : ma5.FB
|  19  | --> pin route of C14         : ra4
|  20  | --> internal route of B1     : ci3.FB
|  21  | --> internal route of B8     : ma4.FB

22 of 22 routes used (utilization = 100.0%)

---------------------
 Routes for block C
---------------------

+-----------------------------------------------+
|input -->           path             : signal |
+-----------------------------------------------+
|   0  | --> pin route of INPUT1      : ale
|   1  | --> internal route of C12    : ma7.FB
|   2  | --> internal route of C13    : ci7.FB
|   3  | --> pin route of INPUT3      : fct1
|   4  | --> pin route of B2          : ra6
|   5  | --> internal route of B1     : ci3.FB
|   6  | --> internal route of A5     : ci1.FB
|   7  | --> internal route of C0     : ma6.FB
|   8  | --> internal route of B11    : ci5.FB
|   9  | --> internal route of D2     : start.FB
|  10  | -->
|  11  | --> pin route of INPUT4      : fct0
|  12  | --> internal route of C15    : ri7.FB
|  13  | --> internal route of B5     : ci4.FB
|  14  | --> internal route of A1     : ci0.FB
|  15  | --> internal route of C1     : ri6.FB
|  16  | --> pin route of A10         : ca6
|  17  | --> pin route of B0          : ca7
|  18  | --> internal route of A11    : ci2.FB
|  19  | --> internal route of C5     : ci6.FB
|  20  | --> pin route of B6          : ra7
|  21  | --> internal route of C7     : cc.FB
```

```
21 of 22 routes used (utilization = 95.5%)

---------------------
 Routes for block D
---------------------
+-----------------------------------------------+
|input -->           path               : signal |
+-----------------------------------------------+
 |   0 | -->
 |   1 | --> internal route of B15      : ri5.FB
 |   2 | -->
 |   3 | --> pin route of A6            : ra9
 |   4 | -->
 |   5 | -->
 |   6 | -->
 |   7 | --> pin route of INPUT4        : fct0
 |   8 | --> pin route of INPUT1        : ale
 |   9 | --> pin route of A0            : ca8
 |  10 | --> pin route of A2            : ra8
 |  11 | --> internal route of A15      : ri2.FB
 |  12 | --> pin route of INPUT0        : ra5
 |  13 | --> pin route of INPUT5        : ra2
 |  14 | --> internal route of D15      : ri9.FB
 |  15 | --> pin route of INPUT3        : fct1
 |  16 | --> internal route of D13      : ci9.FB
 |  17 | --> internal route of D12      : ma9.FB
 |  18 | -->
 |  19 | --> internal route of D5       : ci8.FB
 |  20 | --> internal route of D0       : ma8.FB
 |  21 | --> internal route of D1       : ri8.FB
16 of 22 routes used (utilization = 72.7%)

CHIP route summary: 78 of 88 routes used (utilization = 88.6%)
---------------------------------------------------


Routing Table:

BLOCK        SIGNAL
...D        ra9
...D        ra8
..C.        ra7
..C.        ra6
.B.D        ra5
.B..        ra4
.B..        ra3
A..D        ra2
A...        ra1
A...        ra0
...D        ca8
..C.        ca7
..C.        ca6
.B..        ca5
.B..        ca4
```

```
.B..        ca3
A...        ca2
A...        ca1
A...        ca0
ABCD        ale
ABCD        fct1
ABCD        fct0
....        clk
...D        ma9.FB
...D        ri9.FB
...D        ci9.FB
...D        ma8.FB
...D        ri8.FB
...D        ci8.FB
..C.        ma7.FB
..C.        ri7.FB
..C.        ci7.FB
..C.        ma6.FB
..C.        ri6.FB
..C.        ci6.FB
.B..        ma5.FB
.B.D        ri5.FB
.BC.        ci5.FB
.B..        ma4.FB
.B..        ri4.FB
.BC.        ci4.FB
.B..        ma3.FB
.B..        ri3.FB
.BC.        ci3.FB
A...        ma2.FB
A..D        ri2.FB
ABC.        ci2.FB
A...        ma1.FB
A...        ri1.FB
ABC.        ci1.FB
A...        ma0.FB
A...        ri0.FB
ABC.        ci0.FB
ABC.        start.FB
..C.        cc.FB

Current partitioning requires 78 routes.
----------------------------------------------------
$DEVICE MACH210A fit drama.tt3
$PINS 38 ra9:5 ra8:3 ra7:18 ra6:20 ra5:10 ra4:31 ra3:27 ra2:35 ra1:36
ra0:38 ca8:2 ca7:21 ca6:7 ca5:26 ca4:25 ca3:16 ca2:40 ca1:41 ca0:29 ale:11
fct1:32 fct0:33 clk:13 match3+:39 ma0+:4 ma1+:6 ma2+:8 ma3+:19 ma4+:17
ma5+:15 ma6+:24 ma7+:30 ma8+:43 ma9+:37 match0+:9 match1+:14 match2+:28
start+:42
$NODES 21 ci7:67 ci6:63 cc:64 ci5:58 ci4:55 ci3:53 ci2:50 ci1:47
ci0:45 ci8:71 ci9:75 ri0:46 ri1:48 ri2:52 ri3:54 ri4:56 ri5:60
ri6:61 ri7:68 ri8:69 ri9:76
----------------------------------------------------
```

i overensstemmelse med diagram 10/12-92

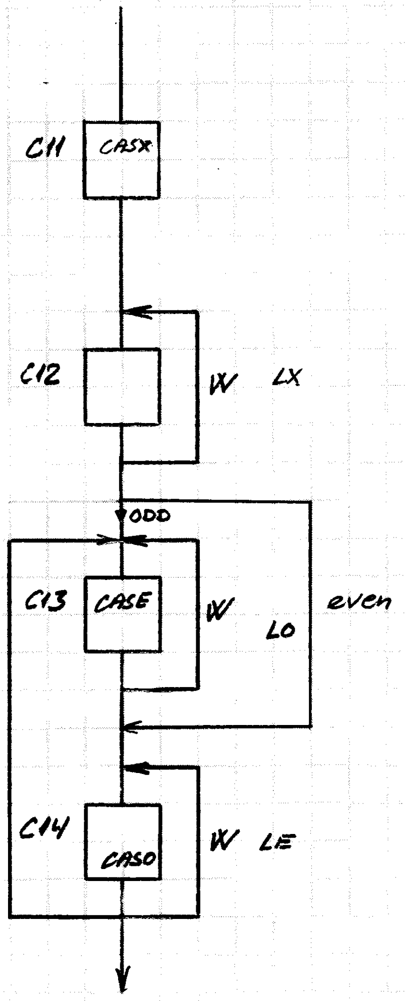
DRAM Controller. DRAMB Main	KAN	Page 1 of 2
	921207	DRAMB



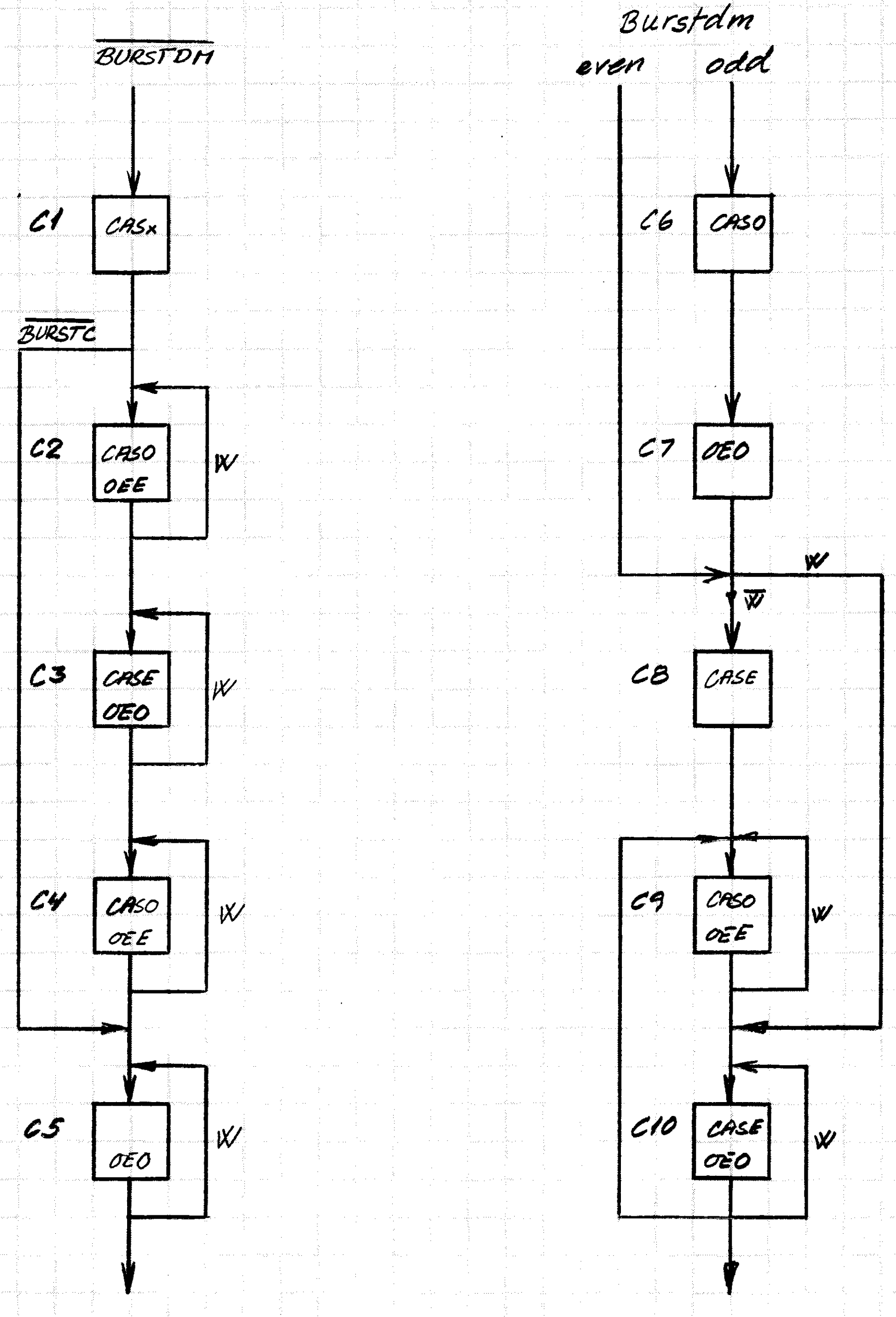
Dansk Data Elektronik A/S

DRAMB Write	Initials KAN	Page
	Date 16/11-1992	Project

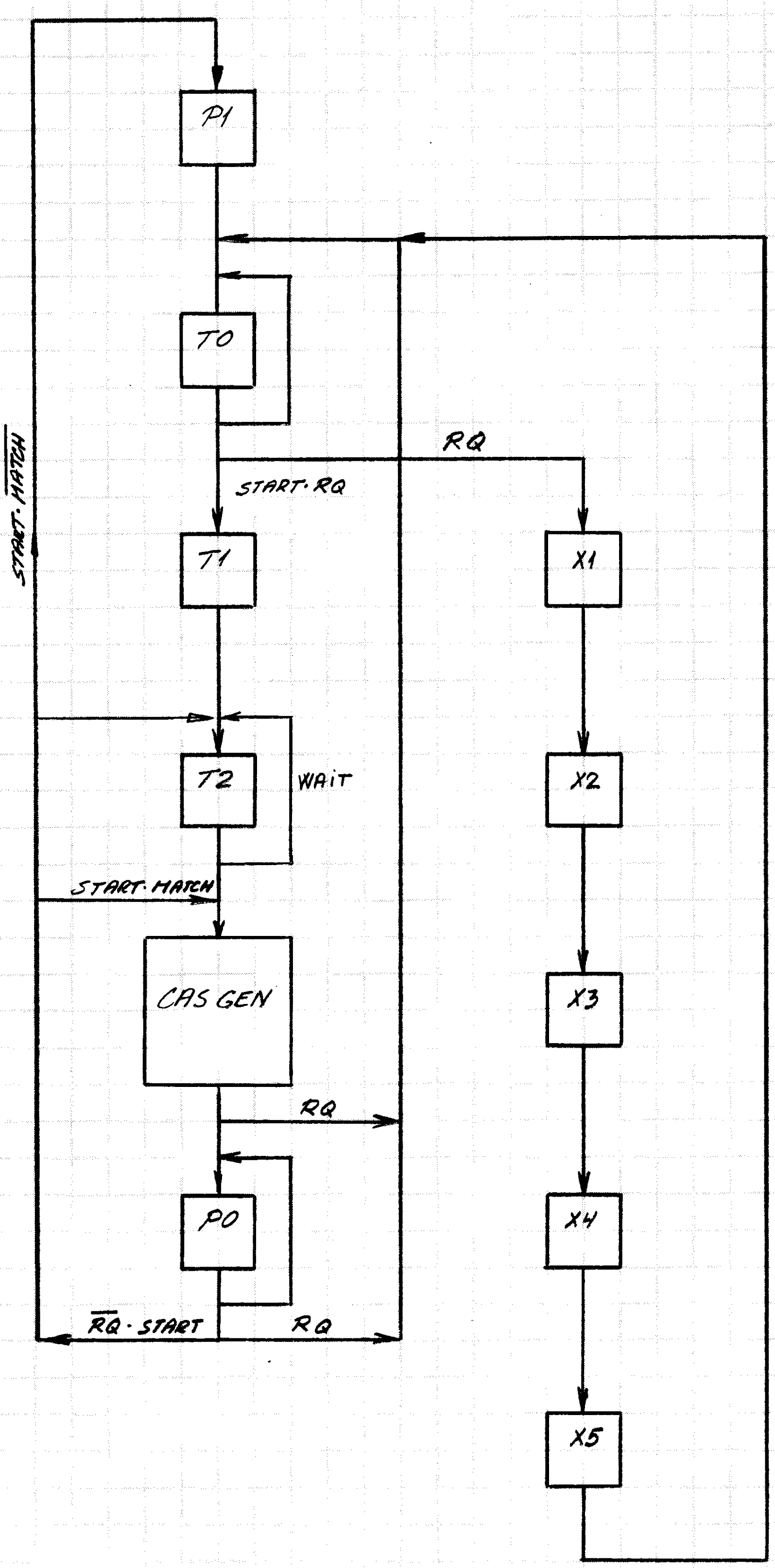
Write



DRAMB read	Initials KAN	Page
	Date 16/11-1992	Project



DRAMB Main	Initials KAN	Page
	Date 12/11-1992	Project

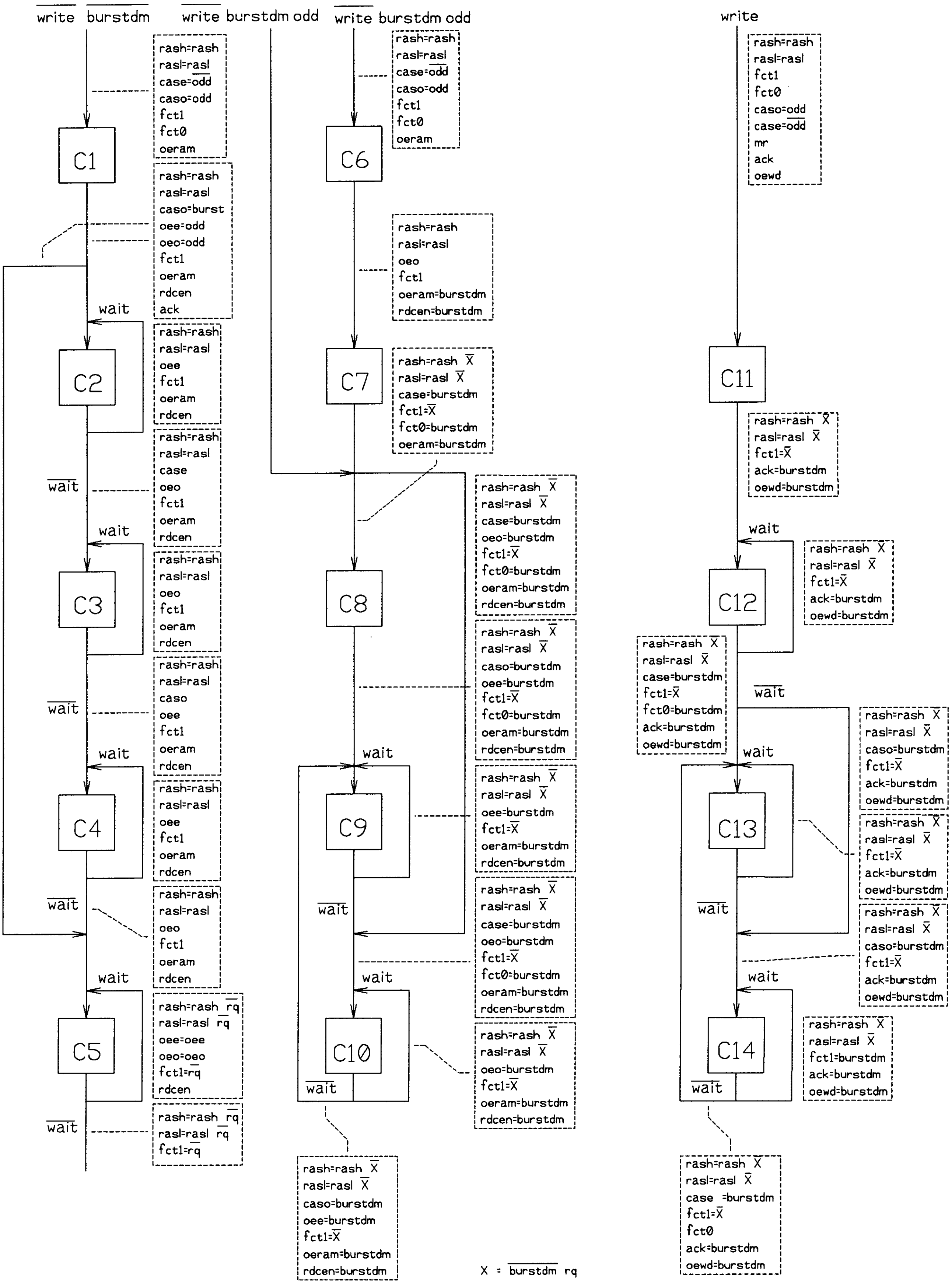

P1
T0
RQ
START·RQ
$\overline{\text{START}\cdot\text{MATCH}}$
T1
X1
T2
WAIT
X2
START·MATCH
CAS GEN
X3
RQ
P0
X4
$\overline{\text{RQ}}$·START
RQ
X5

dde Dansk Data Elektronik A/S

DRAMB state assignment	Initials KAN	Page
	Date 17/11-1992	Project

n4,n3 \ n2n1n0	000	001	011	010	110	111	101	100
0 0	P1			C1	C5	C6	C8	C11
0 1	T2	X2	X3	C2			C7	C12
11	T1	X1	X4	C3			C10	C13
10	P0	T0	X5	C4			C9	C14

DRAM Controller. DRAMB. CAS Gen.	KAN	Page 2 of 2
	921207	DRAMB



$\overline{write}$ $\overline{burstdm}$
$\overline{write}$ burstdm odd
$\overline{write}$ burstdm $\overline{odd}$
write
rash=rash rasl=rasl case=$\overline{odd}$ caso=odd fct1 fct0 oeram
C1
rash=rash rasl=rasl caso=burst oee=odd oeo=odd fct1 oeram rdcen ack
wait
C2
rash=rash rasl=rasl oee fct1 oeram rdcen
$\overline{wait}$
rash=rash rasl=rasl case oeo fct1 oeram rdcen
wait
C3
rash=rash rasl=rasl oeo fct1 oeram rdcen
$\overline{wait}$
rash=rash rasl=rasl caso oee fct1 oeram rdcen
wait
C4
rash=rash rasl=rasl oee fct1 oeram rdcen
$\overline{wait}$
rash=rash rasl=rasl oeo fct1 oeram rdcen
wait
C5
rash=rash $\overline{rq}$ rasl=rasl rq oee=oee oeo=oeo fct1=rq rdcen
$\overline{wait}$
rash=rash $\overline{rq}$ rasl=rasl rq fct1=rq
rash=rash rasl=rasl case=$\overline{odd}$ caso=odd fct1 fct0 oeram
C6
rash=rash rasl=rasl oeo fct1 oeram=burstdm rdcen=burstdm
C7
rash=rash X rasl=rasl X case=burstdm fct1=X fct0=burstdm oeram=burstdm
rash=rash X rasl=rasl X case=burstdm oeo=burstdm fct1=X fct0=burstdm oeram=burstdm rdcen=burstdm
C8
rash=rash X rasl=rasl X caso=burstdm oee=burstdm fct1=X fct0=burstdm oeram=burstdm rdcen=burstdm
wait
C9
rash=rash X rasl=rasl X oee=burstdm fct1=X oeram=burstdm rdcen=burstdm
$\overline{wait}$
rash=rash X rasl=rasl X case=burstdm oeo=burstdm fct1=X fct0=burstdm oeram=burstdm rdcen=burstdm
wait
C10
wait
rash=rash X rasl=rasl X oeo=burstdm fct1=X oeram=burstdm rdcen=burstdm
rash=rash X rasl=rasl X caso=burstdm oee=burstdm fct1=X oeram=burstdm rdcen=burstdm
X = $\overline{burstdm}$ rq
rash=rash rasl=rasl fct1 fct0 caso=odd case=$\overline{odd}$ mr ack oewd
C11
rash=rash X rasl=rasl X fct1=X ack=burstdm oewd=burstdm
wait
C12
rash=rash X rasl=rasl X fct1=X ack=burstdm oewd=burstdm
rash=rash X rasl=rasl X case=burstdm fct1=X fct0=burstdm ack=burstdm oewd=burstdm
$\overline{wait}$
rash=rash X rasl=rasl X caso=burstdm fct1=X ack=burstdm oewd=burstdm
wait
C13
rash=rash X rasl=rasl X fct1=X ack=burstdm oewd=burstdm
$\overline{wait}$
rash=rash X rasl=rasl X caso=burstdm fct1=X ack=burstdm oewd=burstdm
wait
C14
wait
rash=rash X rasl=rasl X fct1=burstdm ack=burstdm oewd=burstdm
rash=rash X rasl=rasl X case =burstdm fct1=X fct0 ack=burstdm oewd=burstdm

```
"Dram controller DRAMB
"Generation of control signals                        kan 921210
"signal mr added                    hvorfor tristates we(3:0) ? KAN 26/7-93
module dramb;

declarations;

drambd device 'mach210a';

"########################################################################
"                        input signals
"########################################################################

x, z, c = .X., .Z., .C.;

"input signals from DRAMA

match3  pin 4;        "compares current ras address with new address
match2  pin 3;        "compares current ras address with new address
match1  pin 2;        "compares current ras address with new address
match0  pin 35;       "compares current ras address with new address

"input signals from processor

ale     pin 36;       "address latch enable from processor

ah      pin 38;       "most significant address bit. Selects rash/rasl
al      pin 39;       "least significant address bit. Selects caso/case

!be3    pin 26;       "byte enable 3 . corresponds to data(31:24)
                      "enables we3 during write cycles.

!be2    pin 9;        "byte enable 2 . corresponds to data(23:16)
                      "enables we2 during write cycles.

!be1    pin 8;        "byte enable 1 . corresponds to data(17: 8)
                      "enables we1 during write cycles.

!be0    pin 6;        "byte enable 0 . corresponds to data( 7: 0)
                      "enables we0 during write cycles.    ændret 5/2-93

!write  pin 11;       " write cycle

!cs     pin 32;       "external address deconding to select DRAM
                      "valid together with start.

!burst  pin 5;        "burst signal from processor. Only significant
                      "during read cycles. burst read cycles will be
                      "four words.

!burstdm pin 33;       "burst signal from DMA device. Both read and write
                      "bursts are performed. DMA bursts end on durstdm.

wait    pin 10;       "delayes DRAM cycles. Data handshake.

ref     pin 7;        "refresh clock. 16 micro seconds cycle time. sync
```

```
clk     pin 13;      "30 ns clock cycle. Inverted version of !SysClk.

"Total number of inputs 17

"#######################################################################
"                           outputs
"#######################################################################

"Output signals to DRAMA

fct1    pin 25   istype 'reg_d,buffer';   "function code to DRAMA
fct0    pin 27   istype 'reg_d,buffer';   "function code to DRAMA

"               fct1 fct0
"                0    0         hold idle; load idle if ale
"                0    1         restart;   load idle if ale
"                1    1         muxcnt;    load page if ale
"                1    0         hold page; load page if ale

"Output to odd address latch

lo      pin 24  istype 'buffer';          "high signal opens latch

"Outputs to DRAM array

rash            pin 20 istype 'reg_d, invert';"ras to expanded memory
rasl            pin 15 istype 'reg_d, invert';"ras to standard memory

cas_e           pin 21 istype 'reg_d, invert';"cas to even bank. al=0
cas_o           pin 14 istype 'reg_d, invert';"cas to odd  bank. al=1

we3     pin 40         istype 'reg_d, invert';"write enable data(31:24)
we2     pin 41         istype 'reg_d, invert';"write enable data(23:16)
we1     pin 42         istype 'reg_d, invert';"write enable data(15: 8)
we0     pin 43         istype 'reg_d, invert';"write enable data( 7: 0)

oeram           pin 29 istype 'reg_d, invert';"common output enable

"Outputs to data path

oee             pin 17 istype 'reg_d, invert';"output enable even
                                                "read data.
oeo             pin 18 istype 'reg_d, invert';"output enable odd
                                                "read data.
oewd    pin 31         istype 'reg_d, buffer';"output enable
                                                "write data.

"outputs to processor notice: sampled on the other clock edge

rdcen           pin 30 istype 'reg_d, invert';"read clock enable.
ack             pin 28 istype 'reg_d, invert';"acknowledge.
mr              pin 16 istype 'reg_d, invert';"memory ready
"mr is generated 1 clock before rdcen.

"total: 15 outputs
```

```
"#########################################################################
"                        internal registers and nodes
"#########################################################################

rx      node 66         istype 'reg_d, buffer';
rq      node 62         istype 'reg_t, buffer';"refresh request

cal     node 73         istype 'reg_d, buffer';

"latched version of cas_o used to generate latch signal to
"odd address: lo.

"signals that must be clocked on ale:

"start is set on ale and cleared in cycle to memory.

start   node 69          istype 'reg_t, buffer';"start memory cycle.
high    node 72          istype 'reg_d, buffer';"ah
odd     node 71          istype 'reg_d, buffer';"al
mh      node 70          istype 'reg_d, buffer';"compares ah with RASx
we              node 61 istype 'reg_d, buffer';"output enable write enable

"state variables

n4              node 51    istype 'reg_d, buffer'; "A13
n3              node 45    istype 'reg_d, buffer'; "A1
n2              node 49    istype 'reg_d, buffer'; "A9
n1              node 48    istype 'reg_d, buffer'; "A7
n0              node 47    istype 'reg_d, buffer'; "A5

"#########################################################################
"                        clock equations
"#########################################################################

equations;

fct1.c = clk; fct0.c = clk;
rash.c = clk; rasl.c = clk; cas_e.c = clk; cas_o.c = clk;
we3.c = clk; we2.c = clk; we1.c = clk; we0.c = clk;
we.c = clk;

oeram.c = clk;
oewd.c  = clk;
rdcen.c = clk;
ack.c   = clk;
mr.c    = clk;

oee.c = clk; oeo.c = clk;

rx.c = clk; rq.c = clk;

cal.le = clk;

high.c = clk; odd.c = clk;
n4.c = clk; n3.c = clk; n2.c = clk; n1.c = clk; n0.c = clk;
```

```
mh.c = clk;
start.c = clk;

rdcen.oe = cs;
ack.oe   = cs;

"########################################################################
"                equations for refresh
"########################################################################

rx.d = ref;
rq.t = !rq.q & !rx.q & ref
     #  rq.q &  n4.q &  n3.q & !n2.q & !n1.q &  n0.q; "state x1

"rq is set by ref and cleared by the refresh cycle.
"refresh cycle is pending when rq = 1

"#########################################################################
"               equations for 'latched' signals
"#########################################################################

high.d = ah  & ale # high.q & !ale;
odd.d  = al  & ale # odd.q  & !ale;

we3.d   = be3 & ale & write # we3.q   & !ale & !(n2.q & !burstdm);
we2.d   = be2 & ale & write # we2.q   & !ale & !(n2.q & !burstdm);
we1.d   = be1 & ale & write # we1.q   & !ale & !(n2.q & !burstdm);
we0.d   = be0 & ale & write # we0.q   & !ale & !(n2.q & !burstdm);

we3.oe  = we.q;
we2.oe  = we.q;
we1.oe  = we.q;
we0.oe  = we.q;
we.d    = n2.q; "more terms added in state diagram;

mh.d   = ah & ale & rash.q # !ah & ale & rasl.q;

start.t = !start.q & ale  "start cycle: start = 1
        #  start.q &  !cs & !ale
        #  start.q & n2.q & !ale; "state c5, c6, c8, c11, a.o.

oeram.d = burstdm & n2.q & n0.q; "to reduce oedram

cal.d = cas_o.q;

lo = cal.q & clk;

"#########################################################################
"               declarations for state diagram
"#########################################################################

declarations;

enner = [n4, n3, n2, n1, n0];
ina = [rq.q, start.q, cs, match3, match2, match1, match0, mh.q,
        write, burst, burstdm, wait, odd.q];
```

```
p1   = ^b00000;
p0   = ^b10000;
t1   = ^b11000;
t2   = ^b01000;
t0   = ^b10001;

x1   = ^b11001;
x2   = ^b01001;
x3   = ^b01011;
x4   = ^b11011;
x5   = ^b10011;

c1   = ^b00010;
c2   = ^b01010;
c3   = ^b11010;
c4   = ^b10010;
c5   = ^b00110;

c6   = ^b00111;
c7   = ^b01101;
c8   = ^b00101;
c9   = ^b10101;
c10  = ^b11101;

c11  = ^b00100;
c12  = ^b01100;
c13  = ^b11100;
c14  = ^b10100;

"#######################################################################
"                       state diagram
"#######################################################################

state_diagram enner;

state t0:
"idle state

rash.d = !rq.q & start.q & cs &  high.q;
rasl.d = !rq.q & start.q & cs & !high.q;

"load idle in t0

fct1.d = !rq.q & start.q & cs; "muxcnt if memory cycle
fct0.d = !rq.q & start.q & cs; "muxcnt if memory cycle

cas_o.d = rq.q;
cas_e.d = rq.q;

"           !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [1, x, x, x, x, x, x, x, x, x, x, x, x]): x1; " refresh
    (ina == [0, 1, 1, x, x, x, x, x, x, x, x, x, x]): t1; "!start cycle
    (ina == [0, 1, 0, x, x, x, x, x, x, x, x, x, x]): t0; "not memory
    (ina == [0, 0, x, x, x, x, x, x, x, x, x, x, x]): t0; "wait
endcase;
```

```
state t1:
"first state in memory cycle

rash.d = rash.q;
rasl.d = rasl.q;

fct1.d = 1;      "hold page
fct0.d = 0;      "hold page

ack.d  = write;
we.d   = 1;
oewd.d = write;

"            !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, x, x, x, x]): t2; "
endcase;

state t2:
"second state in memory cycle

rash.d = rash.q;
rasl.d = rasl.q;

fct1.d = 1;    "mux count except t2
fct0.d = !wait;

cas_o.d = !wait &  odd.q;
cas_e.d = !wait & !odd.q;
oeram.d = !write;

mr.d  = 1;

ack.d = write & wait;
we.d = 1;
oewd.d = write;

"            !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, 1, x, x, 1, x]): t2; "wait for data
    (ina == [x, x, x, x, x, x, x, x, 1, x, 0, 0, x]): c11;"write single
    (ina == [x, x, x, x, x, x, x, x, 1, x, 1, 0, 0]): c11;"write even
    (ina == [x, x, x, x, x, x, x, x, 1, x, 1, 0, 1]): c11;"write odd
    (ina == [x, x, x, x, x, x, x, x, 0, x, x, 1, x]): t2; "read data wait
    (ina == [x, x, x, x, x, x, x, x, 0, x, 0, 0, x]): c1; "read data
    (ina == [x, x, x, x, x, x, x, x, 0, x, 1, 0, 0]): c8; "burst even
    (ina == [x, x, x, x, x, x, x, x, 0, x, 1, 0, 1]): c6; "burst odd
endcase;

state p0:

"page mode idle

rash.d = rash.q & ( !rq.q &  !start.q
# !rq.q & start.q & cs & match3 & match2 & match1 & match0 & mh.q);

rasl.d = rasl.q & ( !rq.q &  !start.q
# !rq.q & start.q & cs & match3 & match2 & match1 & match0 & mh.q);
"idle state

fct1.d = !rq.q & !start.q "p0
       # !rq.q & start.q & cs & match3 & match2 & match1 & match0 & mh.q;

fct0.d =
          !rq.q & start.q &  cs & !match3    "p1
       #  !rq.q & start.q &  cs & !match2    "p1
       #  !rq.q & start.q &  cs & !match1    "p1
       #  !rq.q & start.q &  cs & !match0    "p1
       #  !rq.q & start.q &  cs & !mh.q      "p1
# !wait& !rq.q & start.q & cs & match3 & match2 & match1 & match0 & mh.q;


cas_e.d = !odd.q &
  !wait& !rq.q & start.q & cs & match3 & match2 & match1 & match0 & mh.q;

cas_o.d =  odd.q &
  !wait& !rq.q & start.q & cs & match3 & match2 & match1 & match0 & mh.q;

oeram.d =  !write
       & !rq.q & start.q & cs & match3 & match2 & match1 & match0 & mh.q;
       "t2, t3


ack.d   =  write
       & !rq.q & start.q & cs & match3 & match2 & match1 & match0 & mh.q;

mr.d =   !rq.q & start.q & cs & match3 & match2 & match1 & match0 & mh.q;

we.d = 1;

oewd.d = write;

"           !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [0, 0, x, x, x, x, x, x, x, x, x, x, x]): p0; "page mode idle
    (ina == [1, x, x, x, x, x, x, x, x, x, x, x, x]): t0; "refresh
    (ina == [0, 1, 0, x, x, x, x, x, x, x, x, x, x]): p1; "end page mode
    (ina == [0, 1, 1, 0, x, x, x, x, x, x, x, x, x]): p1; "end page mode
    (ina == [0, 1, 1, x, 0, x, x, x, x, x, x, x, x]): p1; "end page mode
    (ina == [0, 1, 1, x, x, 0, x, x, x, x, x, x, x]): p1; "end page mode
    (ina == [0, 1, 1, x, x, x, 0, x, x, x, x, x, x]): p1; "end page mode
    (ina == [0, 1, 1, x, x, x, x, 0, x, x, x, x, x]): p1; "end page mode
    (ina == [0, 1, 1, 1, 1, 1, 1, 1, 0, x, x, 1, x]): t2; "read wait
    (ina == [0, 1, 1, 1, 1, 1, 1, 1, 0, x, 0, 0, x]): c1; "read
    (ina == [0, 1, 1, 1, 1, 1, 1, 1, 0, x, 1, 0, 0]): c8; "read burst even
    (ina == [0, 1, 1, 1, 1, 1, 1, 1, 0, x, 1, 0, 1]): c6; "read burst odd
    (ina == [0, 1, 1, 1, 1, 1, 1, 1, 1, x, x, 1, x]): t2; "write wait
    (ina == [0, 1, 1, 1, 1, 1, 1, 1, 1, x, 0, 0, x]): c11;"write single
    (ina == [0, 1, 1, 1, 1, 1, 1, 1, 1, x, 1, 0, 0]): c11;"write burst eve
n
    (ina == [0, 1, 1, 1, 1, 1, 1, 1, 1, x, 1, 0, 1]): c11;"write burst odd
endcase;

state p1:
"end page mode
```

```
"            !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, x, x, x, x]): t0;
endcase;


state c1:
"read

rash.d = rash.q;
rasl.d = rasl.q;

cas_o.d = burst;

oee.d = !odd.q;
oeo.d =  odd.q;

fct1.d = 1; "hold page mode
fct0.d = 0;

oeram.d = 1;

rdcen.d = 1;
ack.d   = 1; "acknowledge to 3052

"            !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, 0, x, x, x]): c5; "read single
    (ina == [x, x, x, x, x, x, x, x, x, 1, x, x, x]): c2; "4 word burst
endcase;

state c2:
"4 word read burst
"output enable first word
"cas_o

rash.d = rash.q;
rasl.d = rasl.q;

cas_e.d = !wait;

oee.d =  wait;
oeo.d = !wait;

fct1.d = 1; "hold page
fct0.d = 0;

oeram.d = 1;

rdcen.d = 1;

"            !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, x, x, 1, x]): c2; "wait
    (ina == [x, x, x, x, x, x, x, x, x, x, x, 0, x]): c3; "continue
endcase;

state c3:
```

```
"4 word read burst
"output enable second word
"cas_e

rash.d = rash.q;
rasl.d = rasl.q;

cas_o.d = !wait;

oee.d = !wait;
oeo.d =  wait;

fct1.d = 1; "hold page mode
fct0.d = 0;

oeram.d = 1;

rdcen.d = 1;

"           !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, x, x, 1, x]): c3; "wait
    (ina == [x, x, x, x, x, x, x, x, x, x, x, 0, x]): c4; "continue
endcase;

state c4:
"4 word read burst
"output enable third word
"cas_o

rash.d = rash.q;
rasl.d = rasl.q;

oee.d =  wait;
oeo.d = !wait;

fct1.d = 1; "hold page mode
fct0.d = 0;

oeram.d = 1;

rdcen.d = 1;

"           !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, x, x, 1, x]): c4; "wait
    (ina == [x, x, x, x, x, x, x, x, x, x, x, 0, x]): c5; "continue
endcase;

state c5:
"output enable last word

rash.d = rash.q & !rq.q;
rasl.d = rasl.q & !rq.q;

oee.d =  oee.q & wait;
oeo.d =  oeo.q & wait;
```

```
fctl.d = !rq.q; "if refresh then idle else hold page mode
fct0.d = 0;

rdcen.d = wait;

"            !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, x, x, 1, x]): c5; "wait
    (ina == [0, x, x, x, x, x, x, x, x, x, x, 0, x]): p0; "end
    (ina == [1, x, x, x, x, x, x, x, x, x, x, 0, x]): t0; "end
endcase;

state c6:
"read burst dma odd !starting address

rash.d = rash.q;
rasl.d = rasl.q;

oeo.d = 1;
oee.d = 0;

fctl.d = 1; "hold page mode
fct0.d = 0;

oeram.d = burstdm; "see also equations

rdcen.d = burstdm;

"            !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, x, x, x, x]): c7;
endcase;

state c7:
"read burst dma odd !starting address
"output enable first word

rash.d = rash.q & !(!burstdm & rq.q);
rasl.d = rasl.q & !(!burstdm & rq.q);

cas_e.d = burstdm;

oeo.d =  burstdm & wait;

fctl.d = !(!burstdm & rq.q); "count or idle
fct0.d =    burstdm;

oeram.d = burstdm;

rdcen.d = burstdm & wait;

"            !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, x, 1, 1, x]): c10;"wait
    (ina == [x, x, x, x, x, x, x, x, x, x, 1, 0, x]): c8; "continue
    (ina == [0, x, x, x, x, x, x, x, x, x, 0, x, x]): p0; "end
    (ina == [1, x, x, x, x, x, x, x, x, x, 0, x, x]): t0; "end
endcase;
```

```
state c8:
"read burst dma even !starting address

rash.d = rash.q & !(!burstdm & rq.q);
rasl.d = rasl.q & !(!burstdm & rq.q);

cas_o.d = burstdm;

oee.d = burstdm;

fct1.d = !(!burstdm & rq.q); "hold page mode or idle
fct0.d = 0;

oeram.d = burstdm;

rdcen.d = burstdm;

"            !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, x, 1, x, x]): c9;"wait
    (ina == [0, x, x, x, x, x, x, x, x, x, 0, x, x]): p0; "end
    (ina == [1, x, x, x, x, x, x, x, x, x, 0, x, x]): t0; "end
endcase;

state c9:
"read burst dma even !starting address

rash.d = rash.q & !(!burstdm & rq.q);
rasl.d = rasl.q & !(!burstdm & rq.q);

cas_e.d = burstdm & !wait;

oee.d =   burstdm &  wait;
oeo.d =   burstdm & !wait;

fct1.d = !(!burstdm & rq.q);
fct0.d = burstdm & !wait;  " if wait then hold page mode else count.

oeram.d = burstdm;

rdcen.d = burstdm;

"            !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, x, 1, 1, x]): c9;  "wait
    (ina == [x, x, x, x, x, x, x, x, x, x, 1, 0, x]): c10; "continue
    (ina == [0, x, x, x, x, x, x, x, x, x, 0, x, x]): p0; "end
    (ina == [1, x, x, x, x, x, x, x, x, x, 0, x, x]): t0; "end
endcase;


state c10:
"read burst

rash.d = rash.q & !(!burstdm & rq.q);
rasl.d = rasl.q & !(!burstdm & rq.q);
```

```
cas_o.d = burstdm & !wait;

oee.d =   burstdm & !wait;
oeo.d =   burstdm &  wait;

fct1.d = !(!burstdm & rq.q);
fct0.d = 0;

oeram.d = burstdm;

rdcen.d = burstdm;

"           !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, x, 1, 1, x]): c10; "wait
    (ina == [x, x, x, x, x, x, x, x, x, x, 1, 0, x]): c9;  "continue
    (ina == [0, x, x, x, x, x, x, x, x, x, 0, x, x]): p0; "end
    (ina == [1, x, x, x, x, x, x, x, x, x, 0, x, x]): t0; "end
endcase;


state c11:
"write

rash.d = rash.q & !(!burstdm & rq.q);
rasl.d = rasl.q & !(!burstdm & rq.q);

fct1.d = !(!burstdm & rq.q);
fct0.d = 0;

ack.d = write & burstdm;
oewd.d = burstdm;

"           !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, x, 1, x, x]): c12; "burst write
    (ina == [0, x, x, x, x, x, x, x, x, x, 0, x, x]): p0; "end
    (ina == [1, x, x, x, x, x, x, x, x, x, 0, x, x]): t0; "end
endcase;


state c12:
"write dma burst odd !starting address
"wait for second word

rash.d = rash.q & !(!burstdm & rq.q);
rasl.d = rasl.q & !(!burstdm & rq.q);

cas_e.d = burstdm & !wait &  odd.q;
cas_o.d = burstdm & !wait & !odd.q;

fct1.d = !(!burstdm & rq.q);
fct0.d = burstdm & !wait & odd.q; "if wait then hold page mode else count.

ack.d = write & burstdm;
oewd.d = burstdm;

"           !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
```

```
case(ina == [x, x, x, x, x, x, x, x, x, x, 1, 1, x]): c12; "wait
    (ina == [x, x, x, x, x, x, x, x, x, x, 1, 0, 1]): c13; "continue
    (ina == [x, x, x, x, x, x, x, x, x, x, 1, 0, 0]): c14; "continue
    (ina == [0, x, x, x, x, x, x, x, x, x, 0, x, x]): p0; "end
    (ina == [1, x, x, x, x, x, x, x, x, x, 0, x, x]): t0; "end
endcase;

state c13:
"write dma burst even !starting address

rash.d = rash.q & !(!burstdm & rq.q);
rasl.d = rasl.q & !(!burstdm & rq.q);

cas_o.d = burstdm & !wait;

fct1.d = !(!burstdm & rq.q);
fct0.d = 0;

ack.d = write & burstdm;
oewd.d = burstdm;

"           !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, x, 1, 1, x]): c13; "wait
    (ina == [x, x, x, x, x, x, x, x, x, x, 1, 0, x]): c14; "continue
    (ina == [0, x, x, x, x, x, x, x, x, x, 0, x, x]): p0; "end
    (ina == [1, x, x, x, x, x, x, x, x, x, 0, x, x]): t0; "end
endcase;

state c14:
"write dma burst.

rash.d = rash.q & !(!burstdm & rq.q);
rasl.d = rasl.q & !(!burstdm & rq.q);

cas_e.d = burstdm & !wait;

fct1.d = !(!burstdm & rq.q);
fct0.d = burstdm & !wait;  " if wait then hold page mode else count.

ack.d = write & burstdm;
oewd.d = burstdm;

"           !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
case(ina == [x, x, x, x, x, x, x, x, x, x, 1, 1, x]): c14; "wait
    (ina == [x, x, x, x, x, x, x, x, x, x, 1, 0, x]): c13; "continue
    (ina == [0, x, x, x, x, x, x, x, x, x, 0, x, x]): p0; "end
    (ina == [1, x, x, x, x, x, x, x, x, x, 0, x, x]): t0; "end
endcase;

state x1:
"refresh cycle

rash.d = 1;
rasl.d = 1;

cas_e.d = rq.q;
```

```
   cas_o.d = rq.q;

   "           !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
   case(ina == [x, x, x, x, x, x, x, x, x, x, x, x, x]): x2;
   endcase;

   state x2:
   "refresh cycle

   rash.d = rash.q;
   rasl.d = rasl.q;

   "           !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
   case(ina == [x, x, x, x, x, x, x, x, x, x, x, x, x]): x3;
   endcase;

   state x3:
   "refresh cycle

   rash.d = rash.q;
   rasl.d = rasl.q;

   "           !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
   case(ina == [x, x, x, x, x, x, x, x, x, x, x, x, x]): x4;
   endcase;

   state x4:
   "refresh cycle

   "           !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
   case(ina == [x, x, x, x, x, x, x, x, x, x, x, x, x]): x5;
   endcase;

   state x5:
   "refresh cycle

   "           !rq st cs m3 m2 m1 m0 mh wr bc bd  w  o
   case(ina == [x, x, x, x, x, x, x, x, x, x, x, x, x]): t0;
   endcase;

   test_vectors([clk,ale,cs,write,burst,burstdm,ah,al,be3,be2,be1,be0,
   match3,match2,match1,match0,ref,wait] ->
   [rx,rq,high,odd,mh,we,start,enner,
   !rash,!rasl,!cas_e,!cas_o,!oee,!oeo,
   !we3,!we2,!we1,!we0,!oeram,!rdcen,!ack,fct1,fct0,!mr,oewd,cal,lo]);

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
    [c, x, x, x, x, x, x, x, x, x, x, x, x, x, x, x, x, x] -> "1
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [x, x, x, x, x, x, x,  x, x, x, x, x, x, x, x,x,x,x, x, x, x,x,x,x,x,x,x]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
    [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> "init 2
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, x, x, x, 0, x, 0,  x, x, x, x, x, x, x, x,x,x,x, x, z, z,x,x,x,x,x,x]
;
```

```
  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> "init 3
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, x, x, x, 0, x, 0,  x, x, x, x, x, x, x, x,x,x,x, x, z, z,x,x,x,x,x,x]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "init 4
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, 1, x, x, 0, x, 0,  x, x, x, x, x, x, x, x,x,x,x, x, z, z,x,x,x,x,x,x]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "init 5
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, x, x, x, 0, x, 0,  x, x, x, x, x, x, x, x,x,x,x, x, z, z,x,x,x,x,x,x]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "init 6
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, x, x, x, 0, x, 0,  x, x, x, x, x, x, x, x,x,x,x, x, z, z,x,x,x,x,x,x]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "init 7
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, x, x, x, 0, x, 0,  x, x, x, x, x, x, x, x,x,x,x, x, z, z,x,x,x,x,x,x]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "init 8
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, x, x, x, 0, x, 0,  x, x, x, x, x, x, x, x,x,x,x, x, z, z,x,x,x,x,x,x]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "init 9
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, x, x, x, 0, x, 0,  x, x, x, x, x, x, x, x,x,x,x, x, z, z,x,x,x,x,x,x]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "init 10
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, x, x, x, 0, x, 0,  x, x, x, x, x, x, x, x,x,x,x, x, z, z,x,x,x,x,x,x]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "init 11
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, x, x, x, 0, x, 0,  x, x, x, x, x, x, x, x,x,x,x, x, z, z,x,x,x,x,x,x]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "init 12
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, x, x, x, 0, x, 0,  x, x, x, x, x, x, x, x,x,x,x, x, z, z,x,x,x,x,x,x]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "init 13
```

```
    " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
     [1, x, x, x, 0, x, 0,  x, x, x, x, x, x, x, x,x,x,x, x, z, z,x,x,x,x,x,x]
;

     "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
```

```
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "init 14
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, x, x, x, 0, x, 0,  x, x, x, x, x, x, x, x,x,x,x, x, z, z,x,x,x,x,x,x]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "init 15
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, 0, x, x, 0, x, 0, t0, 0, 0, 0, 0, 0, 0, x,x,x,x, 0, z, z,0,0,0,x,x,x]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> "init 16
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, x, 1, t0, 0, 0, 0, 0, 0, 0, x,x,x,x, 0, z, z,0,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> "init 17
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, x, 0, t0, 0, 0, 0, 0, 0, 0, x,x,x,x, 0, z, z,0,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> "init 18
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, 0, 0, t0, 0, 0, 0, 0, 0, 0, z,z,z,z, 0, z, z,0,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "init 19
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, 1, 0, 0, 0, 0, 1, t0, 0, 0, 0, 0, 0, 0, z,z,z,z, 0, z, z,0,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "refresh 20
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, 1, 0, 0, 0, 0, 1, x1, 0, 0, 1, 1, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,1,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] -> "refresh 21
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, 0, 0, 0, 0, 0, 1, x2, 1, 1, 1, 1, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,1,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> "refresh 22
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, 0, 1, x3, 1, 1, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> "refresh 23
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, 0, 1, x4, 1, 1, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> "refresh 24
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
```

```
    [0, 0, 0, 0, 0, 0, 1, x5, 0, 0, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> " idle   25
```

```
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 0, 1, t0, 0, 0, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> " read   26
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 0, 1, t1, 0, 1, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,1,1,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> " read   27
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 1, 1, t2, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1] -> " read   28
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 1, 1, t2, 0, 1, 0, 0, 0, 0, 0,0,0,0, 1, 0, 0,1,0,1,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> " read   29
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 1, 1, c1, 0, 1, 1, 0, 0, 0, 0,0,0,0, 1, 0, 0,1,1,1,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> " read   30
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 0, 1, c5, 0, 1, 0, 0, 1, 0, z,z,z,z, 1, 1, 1,1,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1] -> " read   31
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 1, 0, c5, 0, 1, 0, 0, 1, 0, 0,0,0,0, 0, 1, 0,1,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> " page m 32
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 1, 0, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0] -> " page m 33
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 1, 1, 1, 1, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, z, z,1,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " page m 34
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 1, 0, 1, 1, c1, 0, 1, 0, 1, 0, 0, 0,0,0,0, 1, 0, 0,1,1,1,0,1,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " page m 35
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 1, 0, 0, 1, c5, 0, 1, 0, 0, 0, 1, z,z,z,z, 1, 1, 1,1,0,0,0,0,0]
```

```
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " page m 36
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c 1
```

```
    [0, 0, 0, 1, 0, 1, 0, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0] -> " page m 37
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 1, 1, 1, p0, 0, 1, 0, 0, 0, 0, 1,1,1,1, 0, z, z,1,0,0,1,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " page m 38
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 1, 1, c11,0, 1, 1, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,1,1,1,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " page m 39
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 1, 0, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0] -> " page m 40
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 1, 1, 1, 1, p0, 0, 1, 0, 0, 0, 0, 1,0,1,0, 0, z, z,1,0,0,1,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " page m 41
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 1, 0, 1, 1, c11,0, 1, 0, 1, 0, 0, 1,0,1,0, 0, 0, 1,1,1,1,1,1,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " page m 42
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 1, 0, 1, 0, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

   "########################################################################
   " four word burst read. Not match3
   "########################################################################

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 43
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 1, 1, 1, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, z, z,1,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0] -> " burstc 44
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 1, 1, p1, 0, 0, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,0,1,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 45
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 0, 1, t0, 0, 0, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 46
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 0, 1, t1, 0, 1, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,1,1,0,0,0,0]
;
```

```
   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 47
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 0, 0, 0, 1, 1, t2, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 48
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 0, 0, 0, 1, 1, c1, 0, 1, 1, 0, 0, 0, 0,0,0,0, 1, 0, 0,1,1,1,0,0,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 49
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 0, 0, 0, 0, 1, c2, 0, 1, 0, 1, 1, 0, z,z,z,z, 1, 1, 1,1,0,0,0,1,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1] -> " burstc 50
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 0, 0, 0, 0, 1, c2, 0, 1, 0, 0, 1, 0, z,z,z,z, 1, 1, 0,1,0,0,0,0,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 51
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 0, 0, 0, 0, 1, c3, 0, 1, 1, 0, 0, 1, z,z,z,z, 1, 1, 0,1,0,0,0,0,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1] -> " burstc 52
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 0, 0, 0, 0, 1, c3, 0, 1, 0, 0, 0, 1, z,z,z,z, 1, 1, 0,1,0,0,0,0,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 53
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 0, 0, 0, 0, 1, c4, 0, 1, 0, 1, 1, 0, z,z,z,z, 1, 1, 0,1,0,0,0,1,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1] -> " burstc 54
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 0, 0, 0, 0, 1, c4, 0, 1, 0, 0, 1, 0, z,z,z,z, 1, 1, 0,1,0,0,0,0,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 55
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 0, 0, 0, 0, 1, c5, 0, 1, 0, 0, 0, 1, z,z,z,z, 1, 1, 0,1,0,0,0,0,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1] -> " burstc 56
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 0, 0, 0, 1, 0, c5, 0, 1, 0, 0, 0, 1, 0,0,0,0, 0, 1, 0,1,0,0,0,0,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 57
```

```
    " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
     [0, 0, 0, 0, 0, 1, 0, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

    "###########################################################################
```

```
"         four word burst read. Start in page mode
"#######################################################################

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 58
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 1, 1, 1, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, z, z,1,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 59
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, 1, 1, c1, 0, 1, 1, 0, 0, 0, 0,0,0,0, 1, 0, 0,1,1,1,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 60
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, 0, 1, c2, 0, 1, 0, 1, 1, 0, z,z,z,z, 1, 1, 1,1,0,0,0,1,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 61
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, 0, 1, c3, 0, 1, 1, 0, 0, 1, z,z,z,z, 1, 1, 0,1,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 62
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, 0, 1, c4, 0, 1, 0, 1, 1, 0, z,z,z,z, 1, 1, 0,1,0,0,0,1,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 63
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, 0, 1, c5, 0, 1, 0, 0, 0, 1, z,z,z,z, 1, 1, 0,1,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> " burstc 64
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, 1, 0, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

 "#######################################################################
 "   burstdm read. Not match 2. Odd starting address. rasl
 "#######################################################################

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 65
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 1, 1, 1, 1, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, z, z,1,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0] -> "burstdm 66
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 1, 0, 1, 1, p1, 0, 0, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,0,1,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
```

```
    [c, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 67
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 1, 0, 0, 1, t0, 0, 0, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,0,0]
;
```

```
     "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
     [c, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 68
    " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
     [0, 0, 0, 1, 0, 0, 1, t1, 0, 1, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,1,1,0,0,0,0]
;

     "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
     [c, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 69
    " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
     [0, 0, 0, 1, 0, 1, 1, t2, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

     "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
     [c, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 70
    " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
     [0, 0, 0, 1, 0, 1, 1, c6, 0, 1, 0, 1, 0, 0, 0,0,0,0, 1, 0, 0,1,1,1,0,1,0]
;

     "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
     [c, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 71
    " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
     [0, 0, 0, 1, 0, 1, 0, c7, 0, 1, 0, 0, 0, 1, 0,0,0,0, 1, 1, 0,1,0,0,0,0,0]
;

     "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
     [c, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 72
    " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
     [0, 0, 0, 1, 0, 1, 0, c8, 0, 1, 1, 0, 0, 0, 0,0,0,0, 1, 0, 0,1,1,0,0,0,0]
;

     "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
     [c, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 73
    " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
     [0, 0, 0, 1, 0, 1, 0, c9, 0, 1, 0, 1, 1, 0, 0,0,0,0, 1, 1, 0,1,0,0,0,1,0]
;

     "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
     [c, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1] -> "burstdm 74
    " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
     [0, 0, 0, 1, 0, 1, 0, c9, 0, 1, 0, 0, 1, 0, 0,0,0,0, 1, 1, 0,1,0,0,0,0,0]
;

     "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
     [c, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 75
    " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
     [0, 0, 0, 1, 0, 1, 0, c10,0, 1, 1, 0, 0, 1, 0,0,0,0, 1, 1, 0,1,1,0,0,0,0]
;

     "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
     [c, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1] -> "burstdm 76
    " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
     [0, 0, 0, 1, 0, 1, 0, c10,0, 1, 0, 0, 0, 1, 0,0,0,0, 1, 1, 0,1,0,0,0,0,0]
;

     "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
     [c, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 77
    " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
     [0, 0, 0, 1, 0, 1, 0, c9, 0, 1, 0, 1, 1, 0, 0,0,0,0, 1, 1, 0,1,0,0,0,1,0]
;

     "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
     [c, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 78
    " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
     [0, 0, 0, 1, 0, 1, 0, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;
```

```
   "######################################################################
   " burst dma read. page mode start. even address. rasl.
   "######################################################################

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 79
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 1, 1, 1, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, z, z,1,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 80
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 1, 1, c8, 0, 1, 1, 0, 0, 0, 0,0,0,0, 1, 0, 0,1,1,1,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 81
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 1, 0, c9, 0, 1, 0, 1, 1, 0, 0,0,0,0, 1, 1, 0,1,0,0,0,1,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 82
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 1, 0, c10,0, 1, 1, 0, 0, 1, 0,0,0,0, 1, 1, 0,1,1,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 83
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 1, 0, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

   "#######################################################################
   " burst DMA read. Not match1. rash. even address
   "#######################################################################

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 84
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 1, 1, 1, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, z, z,1,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0] -> "burstdm 85
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 0, 0, 0, 1, 1, p1, 0, 0, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,0,1,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 86
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 1, 0, 0, 0, 1, t0, 0, 0, 0, 0, 0, 0, z,z,z,z, 0, z, z,0,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 87
   " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
    [0, 0, 1, 0, 0, 0, 1, t1, 1, 0, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,1,1,0,0,0,0]
;
```

```
"  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
[c, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 88
" rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c 1
```

```
  [0, 0, 1, 0, 0, 1, 1, t2, 1, 0, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 89
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 1, 0, 0, 1, 1, c8, 1, 0, 1, 0, 0, 0, 0,0,0,0, 1, 0, 0,1,1,1,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 90
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 1, 0, 0, 1, 0, c9, 1, 0, 0, 1, 1, 0, 0,0,0,0, 1, 1, 0,1,0,0,0,1,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 91
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 1, 0, 0, 1, 0, p0, 1, 0, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

 "######################################################################
 " Burst DMA. Page mode start. Odd address
 "######################################################################

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 92
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 1, 1, 1, 1, 1, p0, 1, 0, 0, 0, 0, 0, 0,0,0,0, 0, z, z,1,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 93
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 1, 1, 0, 1, 1, c6, 1, 0, 0, 1, 0, 0, 0,0,0,0, 1, 0, 0,1,1,1,0,1,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 94
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 1, 1, 0, 1, 0, c7, 1, 0, 0, 0, 0, 1, 0,0,0,0, 1, 1, 0,1,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1] -> "burstdm 95
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 1, 1, 0, 1, 0, c10,1, 0, 1, 0, 0, 1, 0,0,0,0, 1, 1, 0,1,1,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1] -> "burstdm 96
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 1, 1, 0, 1, 0, p0, 1, 0, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

 "#####################################################################
 "Burst DMA write. Not match0. rash. Even address
 "#####################################################################

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0] -> "burstdm 97
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
```

```
    [0, 0, 1, 0, 1, 1, 1, p0, 1, 0, 0, 0, 0, 0, 1,1,1,1, 0, z, z,1,0,0,1,0,0]
;
    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
```

```
      [c, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0] -> "burstdm 98
     " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
      [0, 0, 1, 0, 0, 1, 1, p1, 0, 0, 0, 0, 0, 0, 1,1,1,1, 0, 0, 0,0,1,0,1,0,0]
;

      "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
      [c, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0] -> "burstdm 99
     " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
      [0, 0, 1, 0, 0, 0, 1, t0, 0, 0, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,0,0]
;

      "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
      [c, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0] -> "burstdm 100
     " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
      [0, 0, 1, 0, 0, 0, 1, t1, 1, 0, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,1,1,0,0,0,0]
;

      "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
      [c, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0] -> "burstdm 1
     " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
      [0, 0, 1, 0, 0, 1, 1, t2, 1, 0, 0, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,0,0,1,0,0]
;

      "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
      [c, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0] -> "burstdm 2
     " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
      [0, 0, 1, 0, 0, 1, 1, c11,1, 0, 1, 0, 0, 0, 1,1,1,1, 0, 0, 0,1,1,1,1,0,0]
;

      "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
      [c, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0] -> "burstdm 3
     " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
      [0, 0, 1, 0, 0, 1, 0, c12,1, 0, 0, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,0,0,1,0,0]
;

      "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
      [c, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1] -> "burstdm 4
     " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
      [0, 0, 1, 0, 0, 1, 0, c12,1, 0, 0, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,0,0,1,0,0]
;

      "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
      [c, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0] -> "burstdm 5
     " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
      [0, 0, 1, 0, 0, 1, 0, c14,1, 0, 0, 1, 0, 0, 1,1,1,1, 0, 0, 1,1,0,0,1,1,0]
;

      "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
      [c, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1] -> "burstdm 6
     " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
      [0, 0, 1, 0, 0, 1, 0, c14,1, 0, 0, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,0,0,1,0,0]
;

      "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
      [c, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0] -> "burstdm 7
     " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
      [0, 0, 1, 0, 0, 1, 0, c13,1, 0, 1, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,1,0,1,0,0]
;

      "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
      [c, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1] -> "burstdm 8
     " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
```

```
    [0, 0, 1, 0, 0, 1, 0, c13,1, 0, 0, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,0,0,1,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0] -> "burstdm 9
```

```
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 1, 0, 0, 1, 0, p0, 1, 0, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

  "####################################################################
  "burst DMA write. Page mode !start. rash. Odd address
  "####################################################################

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0] -> "burstdm 10
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 1, 1, 1, 1, 1, p0, 1, 0, 0, 0, 0, 0, 1,1,1,1, 0, 0, 0,1,0,0,1,0,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 11
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 1, 1, 0, 1, 1, c11,1, 0, 0, 1, 0, 0, 1,1,1,1, 0, 0, 1,1,1,1,1,1,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 12
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 1, 1, 0, 1, 0, c12,1, 0, 0, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,0,0,1,0,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 13
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 1, 1, 0, 1, 0, c13,1, 0, 1, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,1,0,1,0,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1] -> "burstdm 14
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 1, 1, 0, 1, 0, c13,1, 0, 0, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,0,0,1,0,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 15
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 1, 1, 0, 1, 0, c14,1, 0, 0, 1, 0, 0, 1,1,1,1, 0, 0, 1,1,0,0,1,1,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 16
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 1, 1, 0, 1, 0, c13,1, 0, 1, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,1,0,1,0,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 17
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 1, 1, 0, 1, 0, p0, 1, 0, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

  "#####################################################################
  " Burst DMA write. rasl. Odd address
  "#####################################################################

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0] -> "burstdm 18
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 0, 1, 0, 1, 1, p0, 1, 0, 0, 0, 0, 0, 1,1,1,1, 0, z, z,1,0,0,1,0,0]
;
```

```
       "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
       [c, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 19
      " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
       [0, 0, 0, 1, 0, 1, 1, p1, 0, 0, 0, 0, 0, 0, 1,1,1,1, 0, 0, 0,0,1,0,1,0,0]
;

       "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
       [c, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 20
      " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
       [0, 0, 0, 1, 0, 0, 1, t0, 0, 0, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,0,0]
;

       "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
       [c, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 21
      " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
       [0, 0, 0, 1, 0, 0, 1, t1, 0, 1, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,1,1,0,0,0,0]
;

       "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
       [c, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 22
      " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
       [0, 0, 0, 1, 0, 1, 1, t2, 0, 1, 0, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,0,0,1,0,0]
;

       "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
       [c, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 23
      " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
       [0, 0, 0, 1, 0, 1, 1, c11,0, 1, 0, 1, 0, 0, 1,1,1,1, 0, 0, 0,1,1,1,1,1,0]
;

       "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
       [c, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 24
      " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
       [0, 0, 0, 1, 0, 1, 0, c12,0, 1, 0, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,0,0,1,0,0]
;

       "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
       [c, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 25
      " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
       [0, 0, 0, 1, 0, 1, 0, c13,0, 1, 1, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,1,0,1,0,0]
;

       "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
       [c, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 26
      " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
       [0, 0, 0, 1, 0, 1, 0, c14,0, 1, 0, 1, 0, 0, 1,1,1,1, 0, 0, 1,1,0,0,1,1,0]
;

       "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
       [c, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 27
      " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
       [0, 0, 0, 1, 0, 1, 0, p0, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,1,0,0,0,0,0]
;

      "######################################################################
      " Burst DMA write. rasl. Page mode start. Even address.
      "######################################################################

       "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf   w
       [c, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0] -> "burstdm 28
      " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
       [0, 0, 0, 0, 1, 1, 1, p0, 0, 1, 0, 0, 0, 0, 1,1,1,1, 0, z, z,1,0,0,1,0,0]
```

```
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0] -> "burstdm 29
```

```
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, 1, 0, 0, 0, 1, 1, c11,0, 1, 1, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,1,1,1,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0] -> "burstdm 30
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [1, 1, 0, 0, 0, 1, 0, c12,0, 1, 0, 0, 0, 0, 1,1,1,1, 0, 0, 1,1,0,0,1,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 31
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 1, 0, 0, 0, 1, 0, c14,0, 1, 0, 1, 0, 0, 1,1,1,1, 0, 0, 1,1,0,0,1,1,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "burstdm 32
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 1, 0, 0, 0, 1, 0, t0, 0, 0, 0, 0, 0, 0, 0,0,0,0, 0, 0, 0,0,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "refresh 33
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 1, 0, 0, 0, 0, 0, x1, 0, 0, 1, 1, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,0,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "refresh 33
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 1, 0, 0, 0, 0, 0, x1, 0, 0, 1, 1, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,1,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "refresh 34
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, 0, 0, x2, 1, 1, 1, 1, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,1,1]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "refresh 34
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, 0, 0, x2, 1, 1, 1, 1, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,1,0]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "refresh 35
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, 0, 0, x3, 1, 1, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,1,1]
;

  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "refresh 35
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
  [0, 0, 0, 0, 0, 0, 0, x3, 1, 1, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,0,0]
;


  "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
  [c, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "refresh 36
 " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
```

```
    [0, 0, 0, 0, 0, 0, 0, x4, 1, 1, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,0,0]
;

    "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
    [c, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "refresh 37
```

```
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 0, 0, 0, 0, 0, x5, 0, 0, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,0,0]
;

   "  le cs wr bc bm ah al b3 b2 b1 b0 m3 m2 m1 m0 rf  w
   [c, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0] -> "refresh 38
  " rx rq hi od mh we st ner rh rl ce co oe oo w[3:0]  or rc ac 1 0 r w c l
   [0, 0, 0, 0, 0, 0, 0, t0, 0, 0, 0, 0, 0, 0, z,z,z,z, 0, 0, 0,0,0,0,0,0,0]
;


  end dramb
```

```
----------------------------------------------------

User pre-assignments:
    fct1                    PIN 25
    fct0                    PIN 27
    rash                    PIN 20
    rasl                    PIN 15
    cas_e                   PIN 21
    cas_o                   PIN 14
    oeram                   PIN 29
    oee                     PIN 17
    oeo                     PIN 18
    rdcen                   PIN 30
    ack                     PIN 28
    mr                      PIN 16
    we                      NODE 61
    n4                      NODE 51
    n3                      NODE 45
    n2                      NODE 49
    n1                      NODE 48
    n0                      NODE 47

---------------------------------------------------
Results of fitting dramb.tt2 into MACH210.
---------------------------------------------------

---------- Dedicated Inputs --------
PIN:10  --> wait
PIN:11  --> write
PIN:32  --> cs
PIN:33  --> burstdm

---------- Clock Inputs ------------
PIN:13  --> clk
PIN:35  --> match0


------------------------------------
      Resources of block A
------------------------------------
Inputs routed to block:  match3 match2 match1 match0 write cs burst burstdm
wait rq.Q n4.Q n3.Q n2.Q n1.Q n0.Q odd.Q start.Q mh.Q

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  A0   (PIN :2   ) -->
      PIN INPUT          : match1
      PRODUCT TERMS      : DOWN
  A1   (NODE:45  ) -->
```

```
      INTERNAL FEEDBACK : n3.D (pts=13)
      PRODUCT TERMS     : LOCAL
  A2   (PIN :3  ) -->
      PIN INPUT         : match2
      PRODUCT TERMS     : UP1
  A3   (NODE:46 ) -->
      PRODUCT TERMS     : UP2
  A4   (PIN :4  ) -->
      PIN INPUT         : match3
      PRODUCT TERMS     : DOWN
  A5   (NODE:47 ) -->
      INTERNAL FEEDBACK : n0.D (pts=12)
      PRODUCT TERMS     : LOCAL
  A6   (PIN :5  ) -->
      PIN INPUT         : burst
      PRODUCT TERMS     : UP1
  A7   (NODE:48 ) -->
      INTERNAL FEEDBACK : n1.D (pts=8)
      PRODUCT TERMS     : LOCAL
  A8   (PIN :6  ) -->
      PIN INPUT         : be0
      PRODUCT TERMS     : UP1
  A9   (NODE:49 ) -->
      INTERNAL FEEDBACK : n2.D (pts=9)
      PRODUCT TERMS     : LOCAL
  A10  (PIN :7  ) -->
      PIN INPUT         : ref
      PRODUCT TERMS     : UP1
  A11  (NODE:50 ) -->
      PRODUCT TERMS     : UP2
  A12  (PIN :8  ) -->
      PIN INPUT         : be1
      PRODUCT TERMS     : DOWN
  A13  (NODE:51 ) -->
      INTERNAL FEEDBACK : n4.D (pts=13)
      PRODUCT TERMS     : LOCAL
  A14  (PIN :9  ) -->
      PIN INPUT         : be2
      PRODUCT TERMS     : UP1
  A15  (NODE:52 ) -->
      PRODUCT TERMS     : UP2

------------------------------------
       Block A resource summary:

                 USED    FREE    UTILIZATION
  product terms  55        9      85.9%
  macro cells    16        0      100.0%
  pins            8        0      100.0%
------------------------------------



------------------------------------
       Resources of block B
------------------------------------
```

```
Inputs routed to block:  match3 match2 match1 match0 cs burst burstdm wait
rq.Q n4.Q n3.Q n2.Q n1.Q n0.Q high.Q odd.Q rash.Q rasl.Q start.Q mh.Q oee.Q
oeo.Q

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  B0   (PIN :21 ) -->
      PIN OUTPUT          : cas_e.D (pts=7)
      PRODUCT TERMS       : LOCAL
  B1   (NODE:53 ) -->
      PRODUCT TERMS       : UP1
  B2   (PIN :20 ) -->
      PIN OUTPUT          : rash.D (pts=11)
      PRODUCT TERMS       : LOCAL
  B3   (NODE:54 ) -->
      PRODUCT TERMS       : UP1
  B4   (PIN :19 ) -->
      PRODUCT TERMS       : UP2
  B5   (NODE:55 ) -->
      PRODUCT TERMS       : DOWN
  B6   (PIN :18 ) -->
      PIN OUTPUT          : oeo.D (pts=8)
      PRODUCT TERMS       : LOCAL
  B7   (NODE:56 ) -->
      PRODUCT TERMS       : DOWN
  B8   (PIN :17 ) -->
      PIN OUTPUT          : oee.D (pts=8)
      PRODUCT TERMS       : LOCAL
  B9   (NODE:57 ) -->
      PRODUCT TERMS       : NOT USED
  B10  (PIN :16 ) -->
      PIN OUTPUT          : mr.D (pts=2)
      PRODUCT TERMS       : LOCAL
  B11  (NODE:58 ) -->
      PRODUCT TERMS       : DOWN
  B12  (PIN :15 ) -->
      PIN OUTPUT          : rasl.D (pts=11)
      PRODUCT TERMS       : LOCAL
  B13  (NODE:59 ) -->
      PRODUCT TERMS       : UP1
  B14  (PIN :14 ) -->
      PIN OUTPUT          : cas_o.D (pts=8)
      PRODUCT TERMS       : LOCAL
  B15  (NODE:60 ) -->
      PRODUCT TERMS       : UP1

------------------------------------
       Block B resource summary:
```

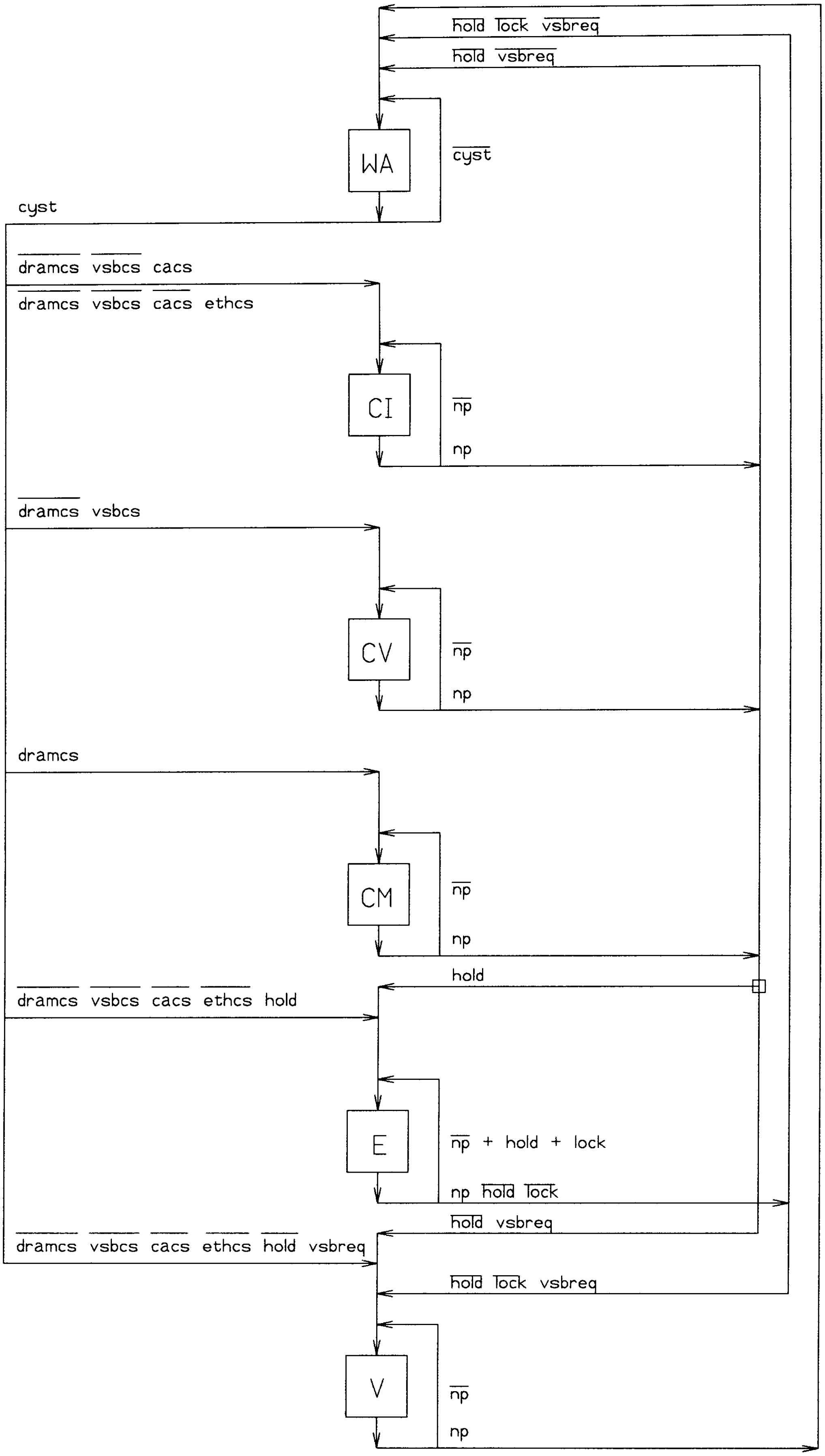
```
                USED     FREE    UTILIZATION
  product terms  55        9       85.9%
  macro cells    15        1       93.8%
  pins            7        1       87.5%
------------------------------------



------------------------------------
      Resources of block C
------------------------------------
Inputs routed to block:  match3 match2 match1 match0 write cs burstdm wait
ref clk rq.Q rx.Q n4.Q n3.Q n2.Q n1.Q n0.Q odd.Q start.Q cal.Q mh.Q

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : cs
   OE2   : <none>

Resource allocation to macro cells:

  C0   (PIN :24 ) -->
      PIN OUTPUT           : lo (pts=1)
      PRODUCT TERMS        : LOCAL
  C1   (NODE:61 ) -->
      INTERNAL FEEDBACK : we.D (pts=3)
      PRODUCT TERMS        : LOCAL
  C2   (PIN :25 ) -->
      PIN OUTPUT           : fct1.D (pts=9)
      PRODUCT TERMS        : LOCAL
  C3   (NODE:62 ) -->
      INTERNAL FEEDBACK : rq.T (pts=2)
      PRODUCT TERMS        : UP1
  C4   (PIN :26 ) -->
      PIN INPUT            : be3
      PRODUCT TERMS        : UP2
  C5   (NODE:63 ) -->
      PRODUCT TERMS        : UP2
  C6   (PIN :27 ) -->
      PIN OUTPUT           : fct0.D (pts=11)
      PRODUCT TERMS        : LOCAL
  C7   (NODE:64 ) -->
      PRODUCT TERMS        : UP1
  C8   (PIN :28 ) -->
      PIN OUTPUT           : ack.D (pts=5)
      PRODUCT TERMS        : UP2
  C9   (NODE:65 ) -->
      PRODUCT TERMS        : UP1
  C10  (PIN :29 ) -->
      PIN OUTPUT           : oeram.D (pts=4)
      PRODUCT TERMS        : UP2
  C11  (NODE:66 ) -->
      INTERNAL FEEDBACK : rx.D (pts=1)
      PRODUCT TERMS        : UP1
  C12  (PIN :30 ) -->
      PIN OUTPUT           : rdcen.D (pts=5)
      PRODUCT TERMS        : UP1
  C13  (NODE:67 ) -->
      PRODUCT TERMS        : UP1
  C14  (PIN :31 ) -->
      PIN OUTPUT           : oewd.D (pts=3)
      PRODUCT TERMS        : UP2
  C15  (NODE:68 ) -->
      PRODUCT TERMS        : UP1

------------------------------------
       Block C resource summary:

                 USED     FREE     UTILIZATION
  product terms  44        20        68.8%
  macro cells    16         0        100.0%
  pins            8         0        100.0%
------------------------------------



------------------------------------
       Resources of block D
------------------------------------
Inputs routed to block:  ale ah al be3 be2 be1 be0 write cs burstdm n2.Q
high.Q odd.Q we3.Q we2.Q we1.Q we0.Q we.Q rash.Q rasl.Q start.Q cas_o.Q

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : we.Q
   OE2   : <none>

Resource allocation to macro cells:

  D0   (PIN :43 ) -->
      PIN OUTPUT           : we0.D (pts=3)
      PRODUCT TERMS        : LOCAL
  D1   (NODE:69 ) -->
      INTERNAL FEEDBACK    : start.T (pts=3)
      PRODUCT TERMS        : LOCAL
  D2   (PIN :42 ) -->
      PIN OUTPUT           : we1.D (pts=3)
      PRODUCT TERMS        : LOCAL
  D3   (NODE:70 ) -->
      INTERNAL FEEDBACK    : mh.D (pts=2)
      PRODUCT TERMS        : LOCAL
  D4   (PIN :41 ) -->
      PIN OUTPUT           : we2.D (pts=3)
      PRODUCT TERMS        : LOCAL
  D5   (NODE:71 ) -->
      INTERNAL FEEDBACK    : odd.D (pts=2)
      PRODUCT TERMS        : LOCAL
  D6   (PIN :40 ) -->
```

```
      PIN OUTPUT          : we3.D (pts=3)
      PRODUCT TERMS       : LOCAL
  D7   (NODE:72 ) -->
      INTERNAL FEEDBACK : high.D (pts=2)
      PRODUCT TERMS       : LOCAL
  D8   (PIN :39 ) -->
      PIN INPUT           : al
      PRODUCT TERMS       : NOT USED
  D9   (NODE:73 ) -->
      INTERNAL FEEDBACK : cal.D (pts=1)
      PRODUCT TERMS       : LOCAL
  D10  (PIN :38 ) -->
      PIN INPUT           : ah
      PRODUCT TERMS       : NOT USED
  D11  (NODE:74 ) -->
      PRODUCT TERMS       : NOT USED
  D12  (PIN :37 ) -->
      PRODUCT TERMS       : NOT USED
  D13  (NODE:75 ) -->
      PRODUCT TERMS       : NOT USED
  D14  (PIN :36 ) -->
      PIN INPUT           : ale
      PRODUCT TERMS       : NOT USED
  D15  (NODE:76 ) -->
      PRODUCT TERMS       : NOT USED

------------------------------------
       Block D resource summary:

                USED     FREE    UTILIZATION
  product terms  22       42       34.4%
  macro cells     9        7       56.3%
  pins            7        1       87.5%
------------------------------------



------------------------------------
       CHIP resource summary:

                USED     FREE    UTILIZATION
  product terms 176       80       68.8%
  macro cells    56        8       87.5%
  pins           36        2       94.7%

------------------------------------

----------------------------------------------------
Routing results.
----------------------------------------------------

---------------------
 Routes for block A
---------------------
+----------------------------------------------+
|input -->            path             : signal |
+----------------------------------------------+
|   0  | --> pin route of INPUT1       : write
|   1  | --> pin route of A2           : match2
|   2  | --> pin route of A4           : match3
|   3  | --> pin route of A6           : burst
|   4  | -->
|   5  | -->
|   6  | -->
|   7  | -->
|   8  | --> internal route of A7      : n1.Q
|   9  | --> pin route of A0           : match1
|  10  | --> pin route of INPUT0       : wait
|  11  | --> pin route of INPUT4       : burstdm
|  12  | --> internal route of D3      : mh.Q
|  13  | --> pin route of INPUT5       : match0
|  14  | --> internal route of A1      : n3.Q
|  15  | --> pin route of INPUT3       : cs
|  16  | --> internal route of A9      : n2.Q
|  17  | --> internal route of C3      : rq.Q
|  18  | --> internal route of A5      : n0.Q
|  19  | --> internal route of D5      : odd.Q
|  20  | --> internal route of A13     : n4.Q
|  21  | --> internal route of D1      : start.Q

18 of 22 routes used (utilization = 81.8%)

---------------------
 Routes for block B
---------------------
+----------------------------------------------+
|input -->             path            : signal |
+----------------------------------------------+
|   0  | --> internal route of D1      : start.Q
|   1  | --> pin route of A2           : match2
|   2  | --> internal route of A1      : n3.Q
|   3  | --> internal route of A9      : n2.Q
|   4  | --> pin route of INPUT4       : burstdm
|   5  | --> internal route of B8      : oee.Q
|   6  | --> internal route of A5      : n0.Q
|   7  | --> internal route of A13     : n4.Q
|   8  | --> pin route of INPUT5       : match0
|   9  | --> internal route of B12     : rasl.Q
|  10  | --> pin route of INPUT0       : wait
|  11  | --> pin route of A0           : match1
|  12  | --> internal route of D3      : mh.Q
|  13  | --> pin route of A4           : match3
|  14  | --> pin route of A6           : burst
|  15  | --> pin route of INPUT3       : cs
|  16  | --> internal route of D7      : high.Q
|  17  | --> internal route of C3      : rq.Q
|  18  | --> internal route of B6      : oeo.Q
|  19  | --> internal route of D5      : odd.Q
|  20  | --> internal route of A7      : n1.Q
|  21  | --> internal route of B2      : rash.Q

22 of 22 routes used (utilization = 100.0%)
```

```
---------------------
 Routes for block C
---------------------
+-----------------------------------------------+
|input -->            path           : signal |
+-----------------------------------------------+
|   0  | --> internal route of D1      : start.Q
|   1  | --> pin route of A2           : match2
|   2  | --> pin route of A4           : match3
|   3  | --> internal route of A9      : n2.Q
|   4  | --> internal route of D5      : odd.Q
|   5  | --> pin route of INPUT5       : match0
|   6  | -->
|   7  | --> internal route of A13     : n4.Q
|   8  | --> pin route of INPUT1       : write
|   9  | --> pin route of A0           : match1
|  10  | --> pin route of INPUT0       : wait
|  11  | --> pin route of INPUT4       : burstdm
|  12  | --> internal route of D3      : mh.Q
|  13  | --> internal route of D9      : cal.Q
|  14  | --> internal route of A1      : n3.Q
|  15  | --> pin route of INPUT3       : cs
|  16  | --> pin route of A10          : ref
|  17  | --> internal route of C3      : rq.Q
|  18  | --> internal route of A5      : n0.Q
|  19  | --> internal route of C11     : rx.Q
|  20  | --> internal route of A7      : n1.Q
|  21  | --> pin route of INPUT2       : clk

21 of 22 routes used (utilization = 95.5%)

---------------------
 Routes for block D
---------------------
+-----------------------------------------------+
|input -->            path           : signal |
+-----------------------------------------------+
|   0  | --> internal route of D1      : start.Q
|   1  | --> internal route of D2      : we1.Q
|   2  | --> pin route of D8           : al
|   3  | --> internal route of A9      : n2.Q
|   4  | --> pin route of A8           : be0
|   5  | --> internal route of D6      : we3.Q
|   6  | --> internal route of D7      : high.Q
|   7  | --> pin route of INPUT4       : burstdm
|   8  | --> internal route of C1      : we.Q
|   9  | --> internal route of B12     : ras1.Q
|  10  | --> internal route of D0      : we0.Q
|  11  | --> pin route of C4           : be3
|  12  | --> pin route of D10          : ah
|  13  | --> internal route of D4      : we2.Q
|  14  | --> internal route of D5      : odd.Q
|  15  | --> pin route of INPUT3       : cs
|  16  | --> pin route of INPUT1       : write
|  17  | --> pin route of D14          : ale
| 18 | --> pin route of A14        : be2
| 19 | --> internal route of B14   : cas_o.Q
| 20 | --> pin route of A12        : be1
| 21 | --> internal route of B2    : rash.Q

22 of 22 routes used (utilization = 100.0%)

CHIP route summary: 83 of 88 routes used (utilization = 94.3%)
---------------------------------------------------


Routing Table:

BLOCK       SIGNAL
ABC.       match3
ABC.       match2
ABC.       match1
ABC.       match0
...D       ale
...D       ah
...D       al
...D       be3
...D       be2
...D       be1
...D       be0
A.CD       write
ABCD       cs
AB..       burst
ABCD       burstdm
ABC.       wait
..C.       ref
..C.       clk
ABC.       rq.Q
..C.       rx.Q
ABC.       n4.Q
ABC.       n3.Q
ABCD       n2.Q
ABC.       n1.Q
ABC.       n0.Q
.B.D       high.Q
ABCD       odd.Q
...D       we3.Q
...D       we2.Q
...D       we1.Q
...D       we0.Q
...D       we.Q
.B.D       rash.Q
.B.D       rasl.Q
ABCD       start.Q
...D       cas_o.Q
..C.       cal.Q
ABC.       mh.Q
.B..       oee.Q
.B..       oeo.Q

Current partitioning requires 83 routes.
```

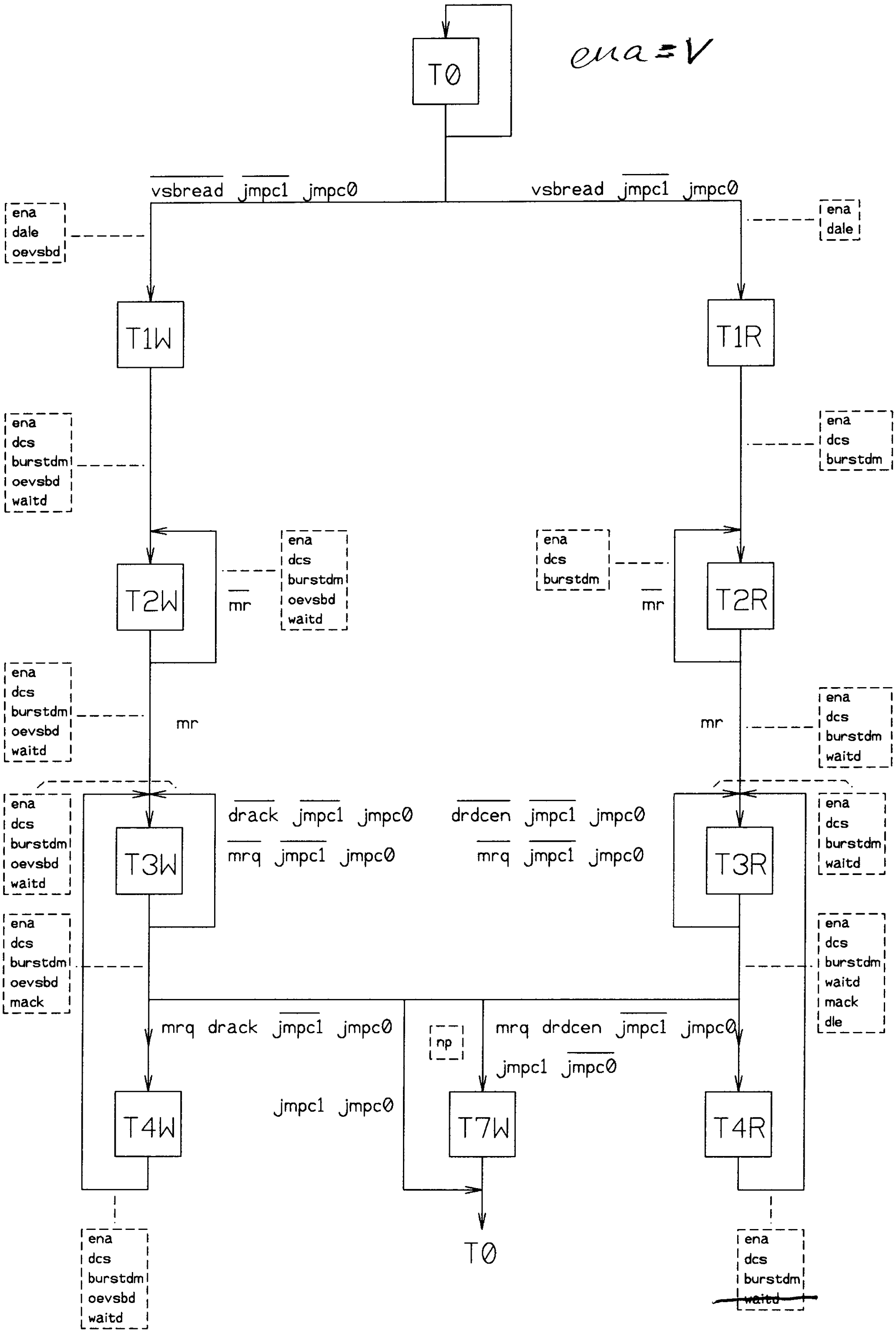
```
----------------------------------------------------
$DEVICE MACH210A fit dramb.tt3
$PINS 36 match3:4 match2:3 match1:2 match0:35 ale:36 ah:38 al:39 be3:26
be2:9 be1:8 be0:6 write:11 cs:32 burst:5 burstdm:33 wait:10 ref:7
clk:13 cas_o-:14 rasl-:15 rash-:20 cas_e-:21 fct0+:27 ack-:28 oeram-:29
fct1+:25 mr-:16 oeo-:18 oee-:17 rdcen-:30 we0-:43 we1-:42 we2-:41 we3-:40
oewd+:31 lo+:24
$NODES 13 rx:66 n2:49 n3:45 n1:48 n0:47 n4:51 rq:62 we:61 start:69
mh:70 odd:71 high:72 cal:73
----------------------------------------------------
```

Central control	KAN	Page 1 of 5
	921213	CC0


$\overline{hold}$ $\overline{lock}$ $\overline{vsbreq}$
$\overline{hold}$ $\overline{vsbreq}$
WA
$\overline{cyst}$
cyst
$\overline{dramcs}$ $\overline{vsbcs}$ cacs
$\overline{dramcs}$ $\overline{vsbcs}$ $\overline{cacs}$ ethcs
CI
$\overline{np}$
np
$\overline{dramcs}$ vsbcs
CV
$\overline{np}$
np
dramcs
CM
$\overline{np}$
np
hold
$\overline{dramcs}$ $\overline{vsbcs}$ $\overline{cacs}$ $\overline{ethcs}$ hold
E
$\overline{np}$ + hold + lock
np $\overline{hold}$ $\overline{lock}$
$\overline{hold}$ vsbreq
$\overline{dramcs}$ $\overline{vsbcs}$ $\overline{cacs}$ $\overline{ethcs}$ $\overline{hold}$ vsbreq
$\overline{hold}$ $\overline{lock}$ vsbreq
V
$\overline{np}$
np

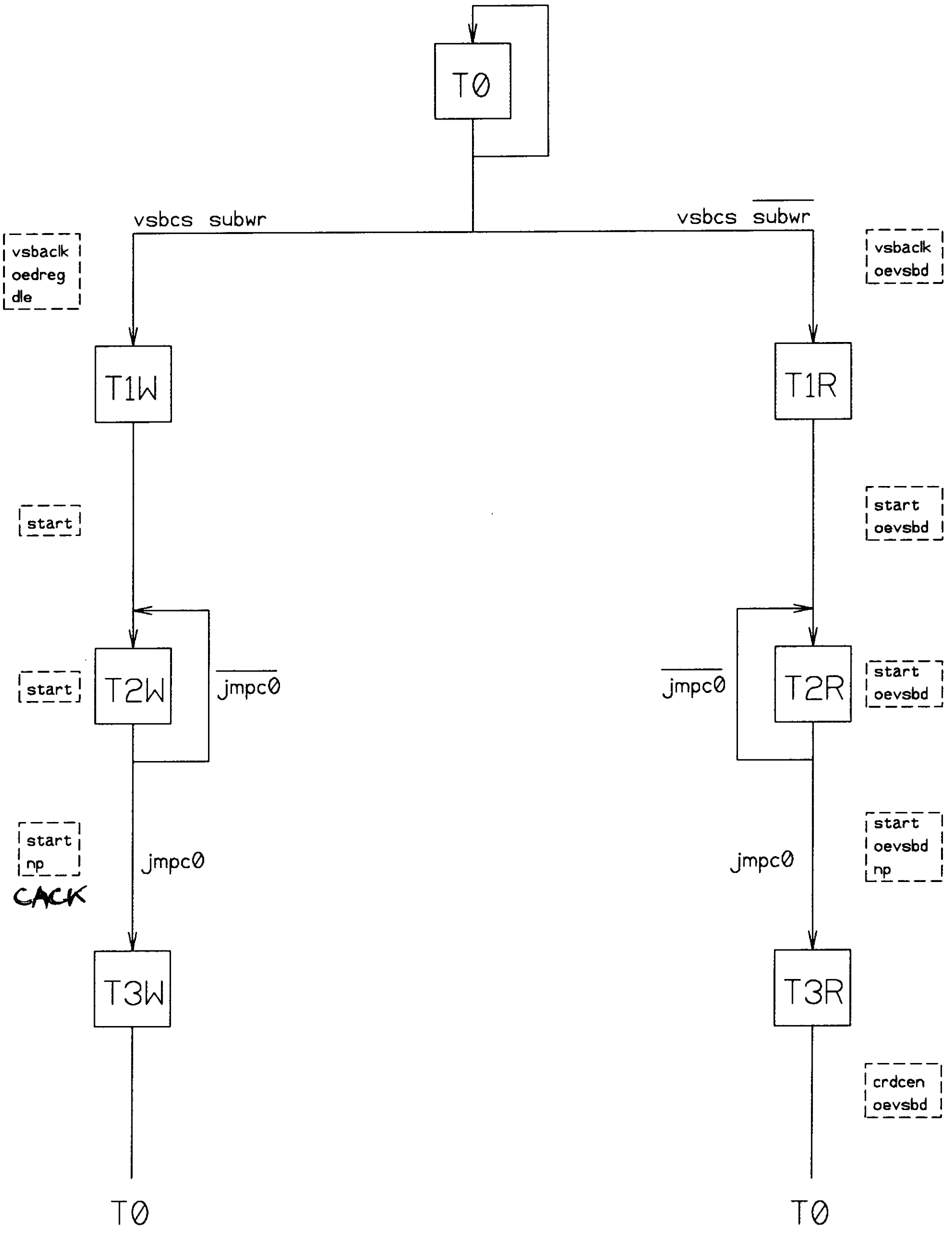
Central control VSB access to BAIO	KAN	Page 2 of 5
	921217	CC0

ena=V

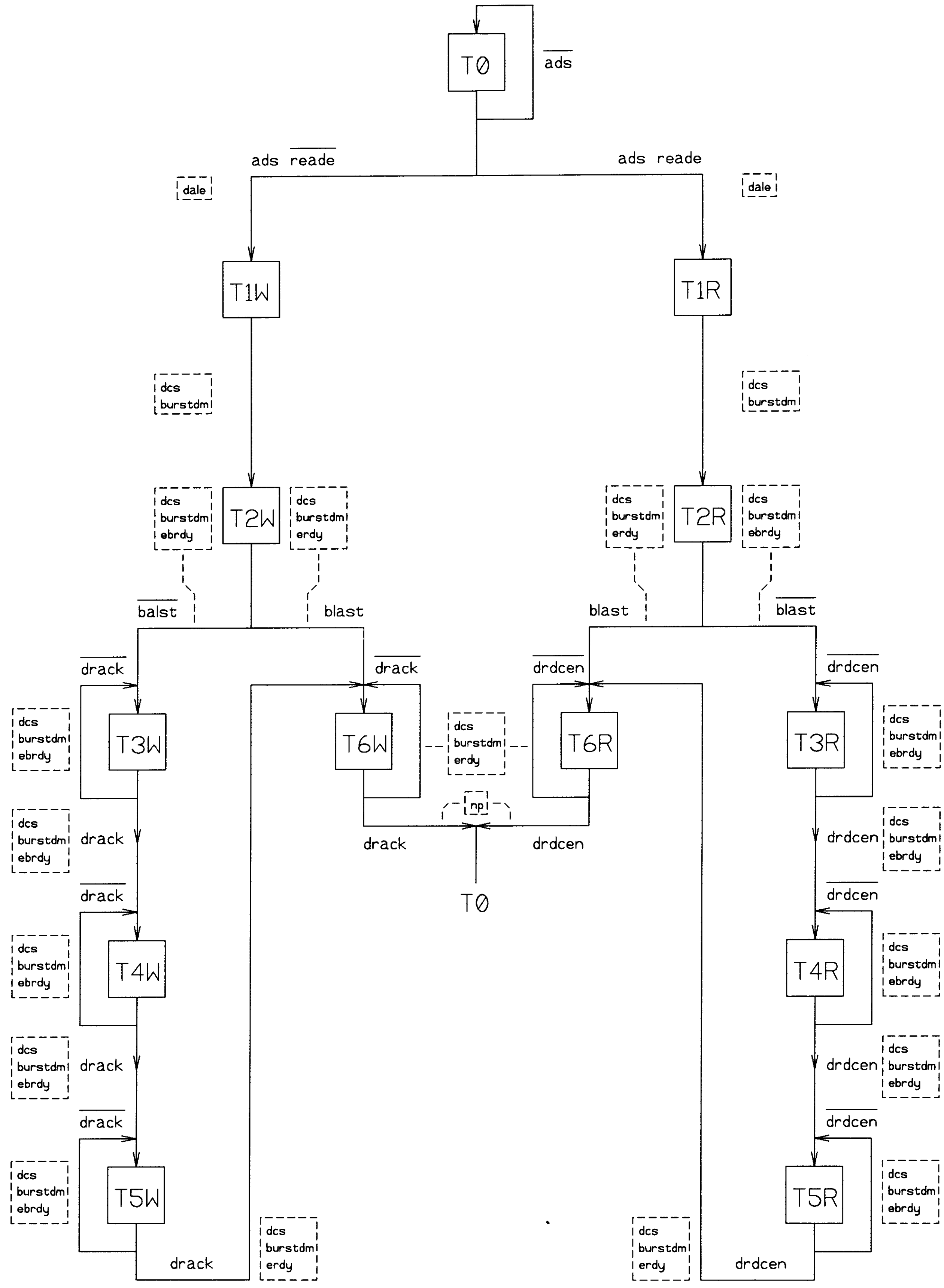


T0
$\overline{vsbread}$ $\overline{jmpc1}$ jmpc0
vsbread $\overline{jmpc1}$ jmpc0
ena dale oevsbd
ena dale
T1W
T1R
ena dcs burstdm oevsbd waitd
ena dcs burstdm
T2W
$\overline{mr}$
ena dcs burstdm oevsbd waitd
ena dcs burstdm
$\overline{mr}$
T2R
ena dcs burstdm oevsbd waitd
mr
mr
ena dcs burstdm waitd
ena dcs burstdm oevsbd waitd
T3W
$\overline{drack}$ $\overline{jmpc1}$ jmpc0
$\overline{mrq}$ $\overline{jmpc1}$ jmpc0
$\overline{drdcen}$ $\overline{jmpc1}$ jmpc0
$\overline{mrq}$ $\overline{jmpc1}$ jmpc0
T3R
ena dcs burstdm waitd
ena dcs burstdm oevsbd mack
ena dcs burstdm waitd mack dle
mrq drack $\overline{jmpc1}$ jmpc0
np
mrq drdcen $\overline{jmpc1}$ jmpc0
jmpc1 $\overline{jmpc0}$
jmpc1 jmpc0
T4W
T7W
T4R
T0
ena dcs burstdm oevsbd waitd
ena dcs burstdm ~~waitd~~

Central control CPU access to vsb	KAN	Page 3 of 5
	921217	CC0



Central control Ethernet access to data memory	KAN	Page 4 of 5
	921219	CC0



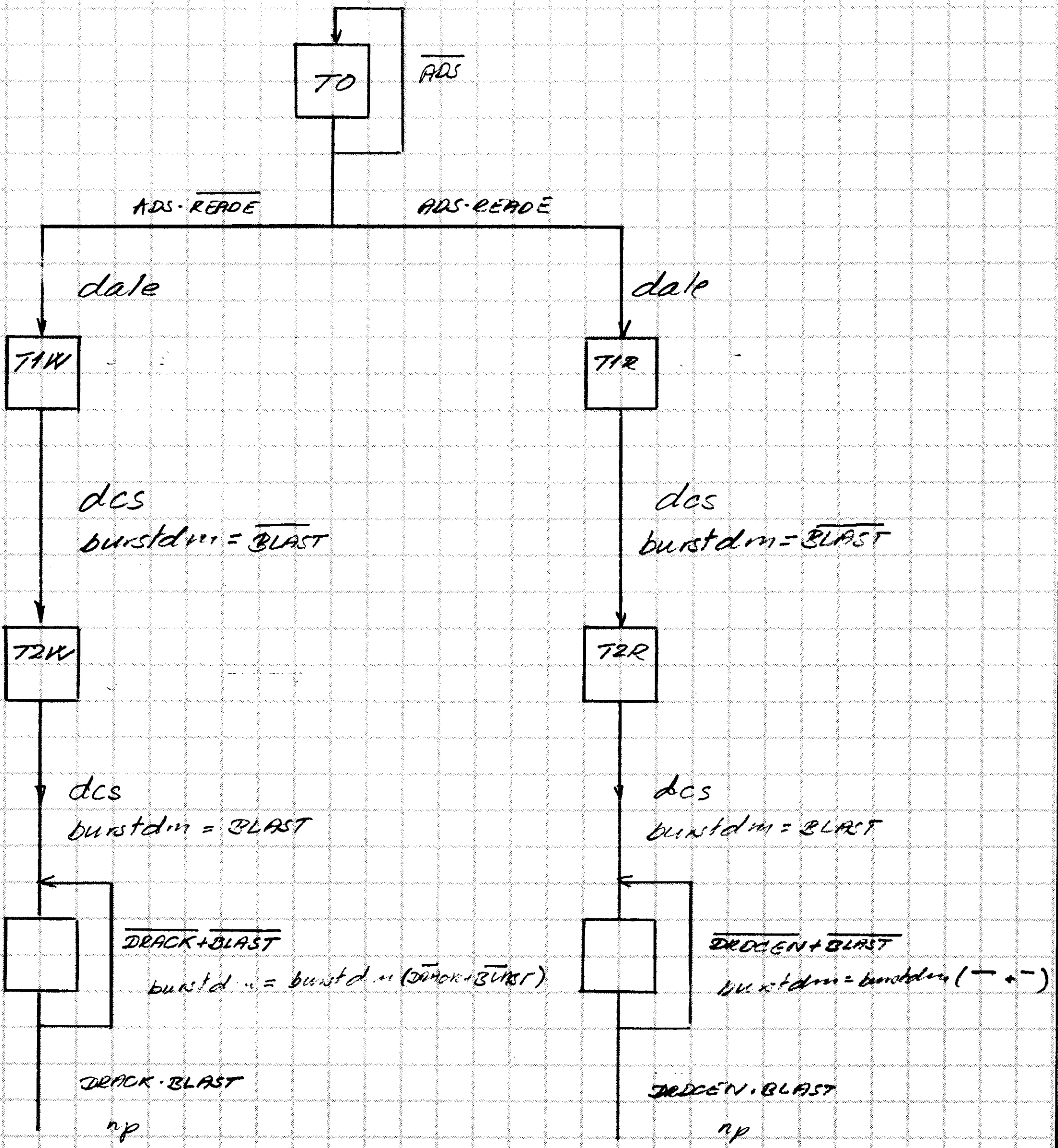
Ethernet access to memory

Initials KAN

Page 4

Date 27/8-93

Project


T0
$\overline{ADS}$
ADS·$\overline{READE}$
ADS·READE
dale
dale
T1W
T1R
dcs
burstdm = $\overline{BLAST}$
dcs
burstdm = $\overline{BLAST}$
T2W
T2R
dcs
burstdm = BLAST
dcs
burstdm = BLAST
$\overline{DRACK}$+$\overline{BLAST}$
burstd.. = burstd.. ($\overline{DRACK}$+$\overline{BLAST}$)
$\overline{DRDCEN}$+$\overline{BLAST}$
burstdm = burstdm ($\overline{\ }$ + $\overline{\ }$)
DRACK·BLAST
np
DRDCEN·BLAST
np

Dansk Data Elektronik A/S

Central control

Initials KAN

Page

Date 22/7-93

Project BA10

Når CPU skriver i datalager får den acknowledge inden skrivningen er udført i lageret.

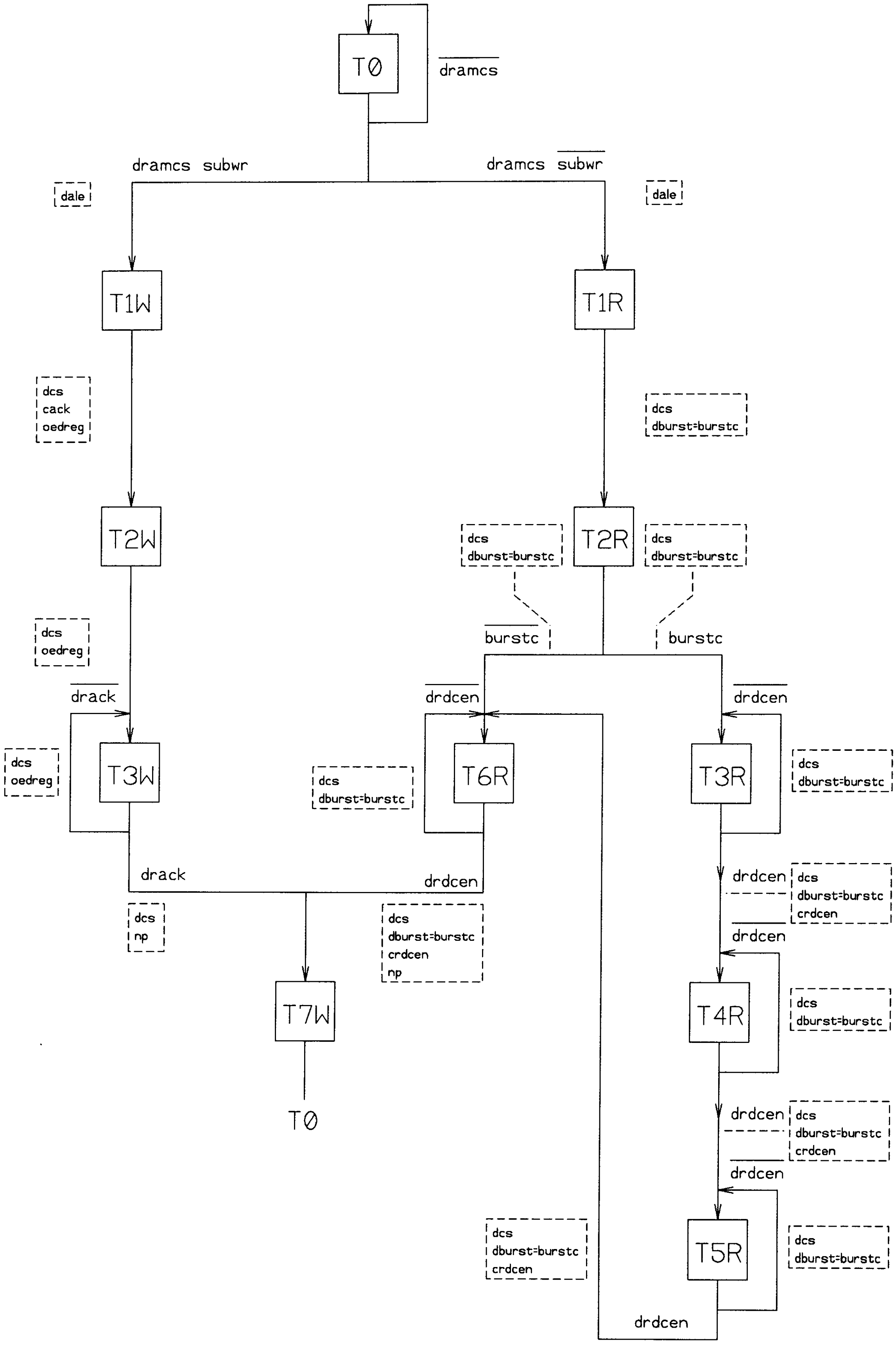
Hvis lager udfører en refreshcycle inden skrivningen, så kan CPU'en sikkert nå at starte en ny skrivning til datalageret og klokke nye data ind i DREG.

Problemet kan løses på to måder:

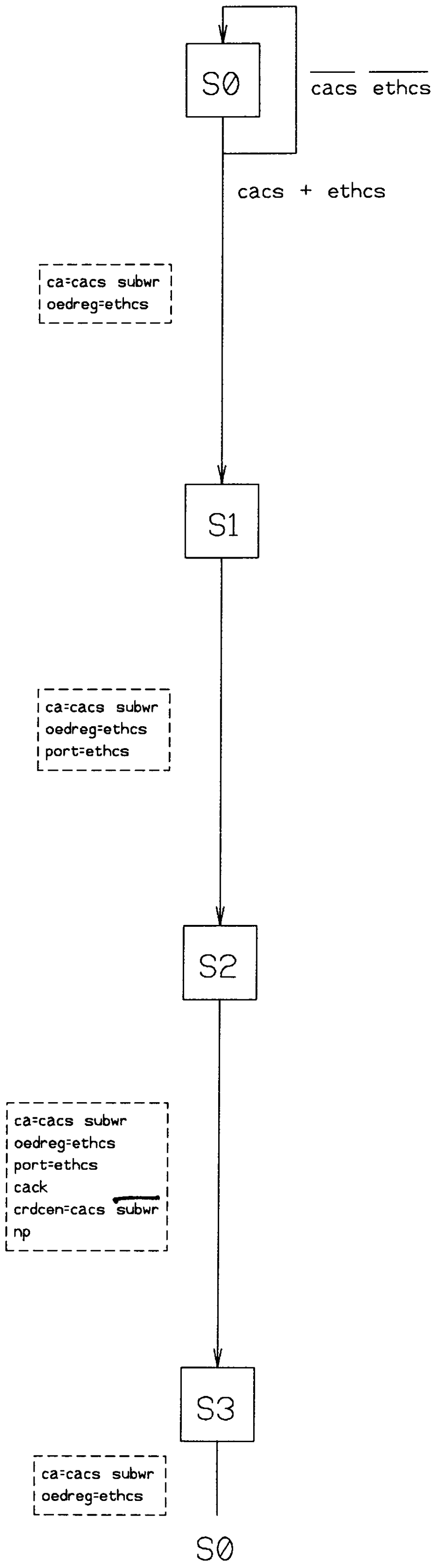
1) postede skrivninger opgives. Central control sender først acknowledge til CPU når skrivningen er udført eller næsten udført.

2) Central control laver et klock signal til skrivning af skrivdata i DREG. Samtidigt med at skrivdata klokkes i DREG sender Central control acknowledge til CPU. CPU er nu frigjort og kan starte en ny skrivning til datalageret d.v.s latche en ny adresse i DREG på ALE (den tidligere adresse er i RAM lageret; DRLE), CPU vil stå med nye skrivdata på output af DREG. Når Central Control vil udføre den nye skrivning klokke den skrivdata i DREG, sender ack til CPU og udfører lagercyklen. Der må gerne komme en cycle fra en anden enhed mellem de to CPU skrivninger.

Central control CPU access to data memory	KAN	Page 5 of 5
	921219	CC0



Central control CPU access to I/O	KAN	Page 6 of 6
	921219	CC0



```
---------------------------------------------------

User pre-assignments:

--------------------------------------------------
Results of fitting cc0.tt2 into MACH220.
--------------------------------------------------

---------- Dedicated Inputs --------
PIN:17  --> jmpc1
PIN:20  --> vsbcs
PIN:51  --> subwr
PIN:54  --> res

---------- Clock Inputs ------------
PIN:15  --> clk
PIN:16  --> cacs
PIN:49  --> drdcen
PIN:50  --> drack


------------------------------------
      Resources of block A
------------------------------------
Inputs routed to block:  res jmpc1 jmpc0 drdcen drack mr dramcs vsbcs cacs
subwr burstc ads blast np v.FB cv.FB nr.FB oeareg.Q ci.FB cm.FB e.FB n2.FB
n1.FB n0.FB i1.FB i0.FB

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : !np oeareg.Q
   OE2   : <none>

Resource allocation to macro cells:

  A0   (PIN :2   ) -->
      PIN OUTPUT          : crdcen.REG (pts=4)
      PRODUCT TERMS       : LOCAL
  A1   (NODE:69 ) -->
      PRODUCT TERMS       : DOWN
  A2   (PIN :3   ) -->
      PIN OUTPUT          : cack.REG (pts=2)
      PRODUCT TERMS       : DOWN
  A3   (NODE:70 ) -->
      INTERNAL FEEDBACK : n0.REG (pts=15)
      PRODUCT TERMS       : LOCAL
  A4   (PIN :4   ) -->
      PIN INPUT           : dramcs
      PRODUCT TERMS       : UP1
  A5   (NODE:71 ) -->
      INTERNAL FEEDBACK : np.REG (pts=7)
      PRODUCT TERMS       : UP2
  A6   (PIN :5   ) -->
      PIN INPUT           : jmpc0
      PRODUCT TERMS      : UP1
  A7   (NODE:72 ) -->
      PRODUCT TERMS      : UP2
  A8   (PIN :6   ) -->
      PIN INPUT          : ethcs
      PRODUCT TERMS      : DOWN
  A9   (NODE:73 ) -->
      INTERNAL FEEDBACK : n1.REG (pts=14)
      PRODUCT TERMS      : LOCAL
  A10  (PIN :7   ) -->
      PIN INPUT          : mrq
      PRODUCT TERMS      : UP1
  A11  (NODE:74 ) -->
      PRODUCT TERMS      : UP2

------------------------------------
       Block A resource summary:

                  USED    FREE    UTILIZATION
  product terms  42        6       87.5%
  macro cells    12        0       100.0%
  pins            6        0       100.0%
------------------------------------



------------------------------------
       Resources of block B
------------------------------------
Inputs routed to block:  res mrq jmpc1 jmpc0 suba5 suba4 suba3 suba2 drdcen
drack burstc blast v.FB nr.FB cm.FB e.FB n2.FB n1.FB n0.FB wc0.Q wc1.Q
wc2.Q wc3.Q ld.Q

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  B0   (PIN :14 ) -->
      PIN OUTPUT         : ld.T (pts=7)
      PRODUCT TERMS      : LOCAL
  B1   (NODE:75 ) -->
      PRODUCT TERMS      : UP1
  B2   (PIN :13 ) -->
      PIN INPUT          : blast
      PRODUCT TERMS      : DOWN
  B3   (NODE:76 ) -->
      INTERNAL FEEDBACK : n2.REG (pts=14)
      PRODUCT TERMS      : LOCAL
  B4   (PIN :12 ) -->
      PIN INPUT          : lock
      PRODUCT TERMS      : UP1
  B5   (NODE:77 ) -->
      INTERNAL FEEDBACK : wc3.T (pts=4)
      PRODUCT TERMS     : UP2
  B6   (PIN :11 ) -->
      PIN OUTPUT        : erdy.REG (pts=4)
      PRODUCT TERMS     : LOCAL
  B7   (NODE:78 ) -->
      INTERNAL FEEDBACK : wc2.T (pts=4)
      PRODUCT TERMS     : UP2
  B8   (PIN :10 ) -->
      PIN INPUT         : reade
      PRODUCT TERMS     : UP1
  B9   (NODE:79 ) -->
      INTERNAL FEEDBACK : wc1.T (pts=4)
      PRODUCT TERMS     : LOCAL
  B10  (PIN :9  ) -->
      PIN INPUT         : vsbread
      PRODUCT TERMS     : NOT USED
  B11  (NODE:80 ) -->
      INTERNAL FEEDBACK : wc0.T (pts=4)
      PRODUCT TERMS     : LOCAL

------------------------------------
       Block B resource summary:

                 USED    FREE    UTILIZATION
  product terms  41        7      85.4%
  macro cells    11        1      91.7%
  pins            6        0      100.0%
------------------------------------



------------------------------------
       Resources of block C
------------------------------------
Inputs routed to block:  res vsbreq vsbread jmpc1 jmpc0 drdcen dramcs vsbcs
ethcs cacs subwr reade ads hold lock np cyst v.FB cv.FB nr.FB ci.FB cm.FB
e.FB n2.FB n1.FB n0.FB

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  C0   (PIN :21 ) -->
      PIN INPUT         : burstc
      PRODUCT TERMS     : DOWN
  C1   (NODE:81 ) -->
      INTERNAL FEEDBACK : nr.REG (pts=13)
      PRODUCT TERMS     : LOCAL
  C2   (PIN :22 ) -->
```

```
      PIN INPUT          : suba2
      PRODUCT TERMS      : UP1
  C3   (NODE:82 ) -->
      INTERNAL FEEDBACK : v.REG (pts=6)
      PRODUCT TERMS      : UP2
  C4   (PIN :23 ) -->
      PIN INPUT          : suba3
      PRODUCT TERMS      : UP1
  C5   (NODE:83 ) -->
      INTERNAL FEEDBACK : e.REG (pts=7)
      PRODUCT TERMS      : UP2
  C6   (PIN :24 ) -->
      PIN INPUT          : ads
      PRODUCT TERMS      : UP1
  C7   (NODE:84 ) -->
      INTERNAL FEEDBACK : cv.REG (pts=2)
      PRODUCT TERMS      : UP2
  C8   (PIN :25 ) -->
      PIN INPUT          : suba4
      PRODUCT TERMS      : UP1
  C9   (NODE:85 ) -->
      INTERNAL FEEDBACK : cm.REG (pts=2)
      PRODUCT TERMS      : LOCAL
  C10  (PIN :26 ) -->
      PRODUCT TERMS      : NOT USED
  C11  (NODE:86 ) -->
      PRODUCT TERMS      : NOT USED

------------------------------------
       Block C resource summary:

                  USED    FREE    UTILIZATION
  product terms   30      18       62.5%
  macro cells     10       2       83.3%
  pins             5       1       83.3%
------------------------------------



------------------------------------
       Resources of block D
------------------------------------
Inputs routed to block:  res vsbreq drack dramcs vsbcs ethcs cacs subwr
hold lock holda oeareg oedreg np cyst bc v.FB cv.FB nr.FB ci.FB cm.FB e.FB
oevsba.Q n2.FB n1.FB n0.FB

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  D0   (PIN :33 ) -->
```

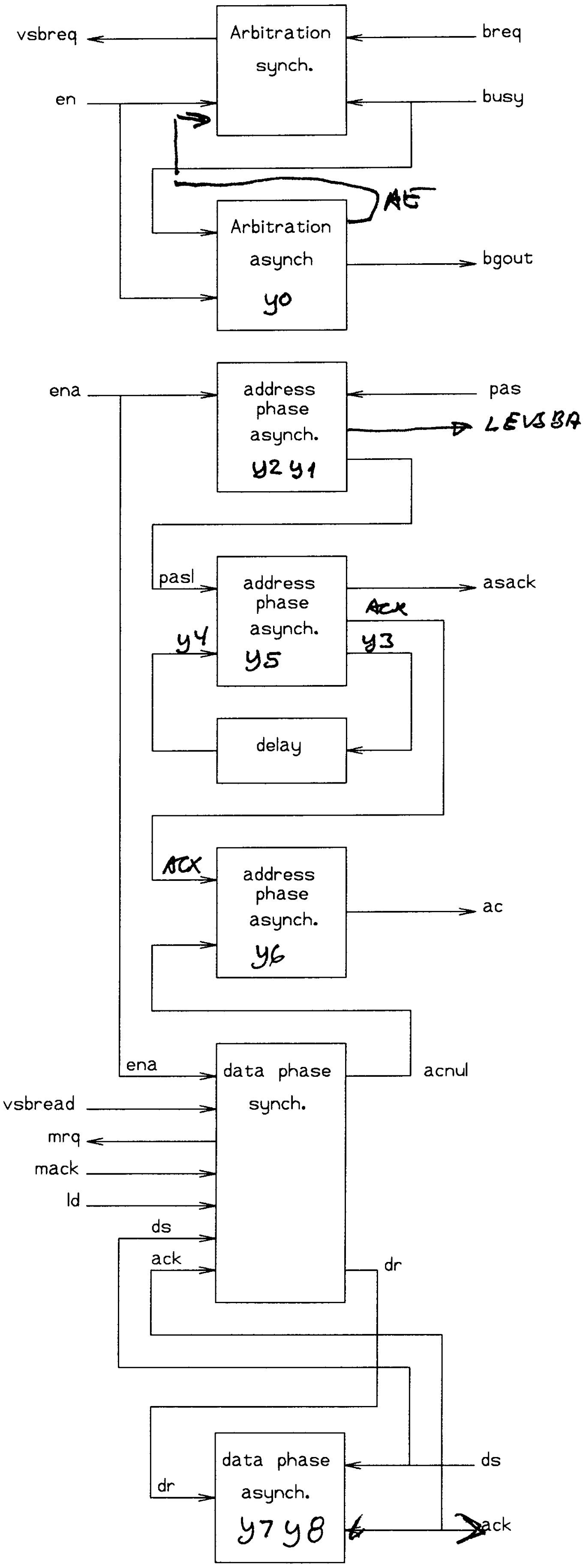
```
      PIN OUTPUT          : oevsba.D (pts=6)
      PRODUCT TERMS       : LOCAL
  D1   (NODE:87 ) -->
      PRODUCT TERMS       : UP1
  D2   (PIN :32 ) -->
      PIN INPUT           : mr
      PRODUCT TERMS       : DOWN
  D3   (NODE:88 ) -->
      INTERNAL FEEDBACK : ci.REG (pts=3)
      PRODUCT TERMS       : DOWN
  D4   (PIN :31 ) -->
      PIN INPUT           : holda
      PIN OUTPUT          : holda.REG (pts=6)
      PRODUCT TERMS       : LOCAL
  D5   (NODE:89 ) -->
      PRODUCT TERMS       : DOWN
  D6   (PIN :30 ) -->
      PIN INPUT           : oeareg
      PIN OUTPUT          : oeareg.REG (pts=5)
      PRODUCT TERMS       : LOCAL
  D7   (NODE:90 ) -->
      INTERNAL FEEDBACK : cyst.REG (pts=4)
      PRODUCT TERMS       : LOCAL
  D8   (PIN :29 ) -->
      PIN INPUT           : suba5
      PRODUCT TERMS       : NOT USED
  D9   (NODE:91 ) -->
      PRODUCT TERMS       : DOWN
  D10  (PIN :28 ) -->
      PIN INPUT           : oedreg
      PIN OUTPUT          : oedreg.REG (pts=5)
      PRODUCT TERMS       : LOCAL
  D11  (NODE:92 ) -->
      PRODUCT TERMS       : NOT USED

------------------------------------
       Block D resource summary:

                  USED    FREE    UTILIZATION
  product terms  29       19       60.4%
  macro cells    10        2       83.3%
  pins            6        0       100.0%
------------------------------------



------------------------------------
       Resources of block E
------------------------------------
Inputs routed to block:  res mrq vsbread jmpc1 jmpc0 drdcen drack dramcs
vsbcs subwr ads v.FB cv.FB nr.FB cm.FB e.FB n2.FB n1.FB n0.FB

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1    : <none>
   OE2    : <none>

Resource allocation to macro cells:

  E0    (PIN :36 ) -->
      PIN OUTPUT          : dle.REG (pts=2)
      PRODUCT TERMS       : LOCAL
  E1    (NODE:93 ) -->
      PRODUCT TERMS       : NOT USED
  E2    (PIN :37 ) -->
      PIN OUTPUT          : dale.REG (pts=3)
      PRODUCT TERMS       : LOCAL
  E3    (NODE:94 ) -->
      PRODUCT TERMS       : DOWN
  E4    (PIN :38 ) -->
      PIN OUTPUT          : oevsbd.D (pts=7)
      PRODUCT TERMS       : LOCAL
  E5    (NODE:95 ) -->
      PRODUCT TERMS       : DOWN
  E6    (PIN :39 ) -->
      PIN OUTPUT          : dcs.REG (pts=11)
      PRODUCT TERMS       : LOCAL
  E7    (NODE:96 ) -->
      PRODUCT TERMS       : UP1
  E8    (PIN :40 ) -->
      PIN OUTPUT          : mack.REG (pts=2)
      PRODUCT TERMS       : LOCAL
  E9    (NODE:97 ) -->
      PRODUCT TERMS       : DOWN
  E10   (PIN :41 ) -->
      PIN OUTPUT          : burstdm.REG (pts=7)
      PRODUCT TERMS       : LOCAL
  E11   (NODE:98 ) -->
      PRODUCT TERMS       : NOT USED

------------------------------------
       Block E resource summary:

                  USED    FREE    UTILIZATION
  product terms   32      16       66.7%
  macro cells     10       2       83.3%
  pins             6       0      100.0%
------------------------------------



------------------------------------
       Resources of block F
------------------------------------
Inputs routed to block:  mrq dburst drdcen drack mr burstc blast v.FB nr.FB
cm.FB e.FB n2.FB n1.FB n0.FB

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  F0   (PIN :48 ) -->
      PIN OUTPUT           : ebrdy.REG (pts=4)
      PRODUCT TERMS        : LOCAL
  F1   (NODE:99 ) -->
      PRODUCT TERMS        : DOWN
  F2   (PIN :47 ) -->
      PIN OUTPUT           : waitd.REG (pts=6)
      PRODUCT TERMS        : LOCAL
  F3   (NODE:100) -->
      PRODUCT TERMS        : NOT USED
  F4   (PIN :46 ) -->
      PIN INPUT            : vsbreq
      PRODUCT TERMS        : NOT USED
  F5   (NODE:101) -->
      PRODUCT TERMS        : NOT USED
  F6   (PIN :45 ) -->
      PIN INPUT            : dburst
      PIN OUTPUT           : dburst.REG (pts=3)
      PRODUCT TERMS        : LOCAL
  F7   (NODE:102) -->
      PRODUCT TERMS        : NOT USED
  F8   (PIN :44 ) -->
      PRODUCT TERMS        : NOT USED
  F9   (NODE:103) -->
      PRODUCT TERMS        : NOT USED
  F10  (PIN :43 ) -->
      PRODUCT TERMS        : NOT USED
  F11  (NODE:104) -->
      PRODUCT TERMS        : NOT USED

------------------------------------
       Block F resource summary:

                 USED     FREE    UTILIZATION
  product terms  13        35      27.1%
  macro cells     4         8      33.3%
  pins            4         2      66.7%
------------------------------------



------------------------------------
       Resources of block G
------------------------------------
Inputs routed to block:  res jmpcl vsbcs v.FB cv.FB nr.FB n2.FB n1.FB n0.FB

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1    : <none>
   OE2    : <none>

Resource allocation to macro cells:

  G0   (PIN :55 ) -->
      PIN OUTPUT          : vsbaclk.REG (pts=1)
      PRODUCT TERMS       : LOCAL
  G1   (NODE:105) -->
      PRODUCT TERMS       : NOT USED
  G2   (PIN :56 ) -->
      PIN OUTPUT          : ena.REG (pts=4)
      PRODUCT TERMS       : LOCAL
  G3   (NODE:106) -->
      PRODUCT TERMS       : NOT USED
  G4   (PIN :57 ) -->
      PIN OUTPUT          : start.REG (pts=1)
      PRODUCT TERMS       : LOCAL
  G5   (NODE:107) -->
      PRODUCT TERMS       : NOT USED
  G6   (PIN :58 ) -->
      PIN OUTPUT          : dwr.REG (pts=1)
      PRODUCT TERMS       : LOCAL
  G7   (NODE:108) -->
      PRODUCT TERMS       : NOT USED
  G8   (PIN :59 ) -->
      PIN OUTPUT          : mast.REG (pts=1)
      PRODUCT TERMS       : LOCAL
  G9   (NODE:109) -->
      PRODUCT TERMS       : NOT USED
  G10  (PIN :60 ) -->
      PIN OUTPUT          : en.REG (pts=1)
      PRODUCT TERMS       : LOCAL
  G11  (NODE:110) -->
      PRODUCT TERMS       : NOT USED

------------------------------------
       Block G resource summary:

                  USED     FREE    UTILIZATION
  product terms    9        39       18.8%
  macro cells      6         6       50.0%
  pins             6         0       100.0%
------------------------------------



------------------------------------
       Resources of block H
------------------------------------
Inputs routed to block:  res ethcs cacs subwr cv.FB ci.FB cm.FB i1.FB i0.FB

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1    : <none>
   OE2    : <none>

Resource allocation to macro cells:

  H0   (PIN :67 ) -->
      PIN OUTPUT           : port.REG (pts=1)
      PRODUCT TERMS        : LOCAL
  H1   (NODE:111) -->
      INTERNAL FEEDBACK : i0.REG (pts=2)
      PRODUCT TERMS        : LOCAL
  H2   (PIN :66 ) -->
      PIN OUTPUT           : ca.REG (pts=1)
      PRODUCT TERMS        : LOCAL
  H3   (NODE:112) -->
      INTERNAL FEEDBACK : i1.REG (pts=2)
      PRODUCT TERMS        : LOCAL
  H4   (PIN :65 ) -->
      PRODUCT TERMS        : NOT USED
  H5   (NODE:113) -->
      INTERNAL FEEDBACK : bc.REG (pts=1)
      PRODUCT TERMS        : LOCAL
  H6   (PIN :64 ) -->
      PRODUCT TERMS        : NOT USED
  H7   (NODE:114) -->
      PRODUCT TERMS        : NOT USED
  H8   (PIN :63 ) -->
      PRODUCT TERMS        : NOT USED
  H9   (NODE:115) -->
      PRODUCT TERMS        : NOT USED
  H10  (PIN :62 ) -->
      PIN INPUT            : hold
      PRODUCT TERMS        : NOT USED
  H11  (NODE:116) -->
      PRODUCT TERMS        : NOT USED

------------------------------------
       Block H resource summary:

                  USED    FREE    UTILIZATION
  product terms    7       41       14.6%
  macro cells      5        7       41.7%
  pins             3        3       50.0%
------------------------------------



------------------------------------
       CHIP resource summary:

                  USED    FREE    UTILIZATION
  product terms 203       181       52.9%
  macro cells     68       28       70.8%
  pins            50        6       89.3%

------------------------------------
```

```
----------------------------------------------------
Routing results.
----------------------------------------------------

---------------------
 Routes for block A
---------------------
+-----------------------------------------------+
|input -->           path              : signal |
+-----------------------------------------------+
|   0  | --> pin route of INPUT7       : res
|   1  | --> internal route of C7      : cv.FB
|   2  | --> pin route of A4           : dramcs
|   3  | --> pin route of INPUT2       : jmpc1
|   4  | --> pin route of INPUT3       : vsbcs
|   5  | --> pin route of D2           : mr
|   6  | --> pin route of C0           : burstc
|   7  | --> internal route of D3      : ci.FB
|   8  | --> internal route of H3      : i1.FB
|   9  | --> internal route of A3      : n0.FB
|  10  | --> internal route of A5      : np
|  11  | --> internal route of A9      : n1.FB
|  12  | --> pin route of A6           : jmpc0
|  13  | --> internal route of C9      : cm.FB
|  14  | --> pin route of B2           : blast
|  15  | --> internal route of B3      : n2.FB
|  16  | --> internal route of C1      : nr.FB
|  17  | --> internal route of C3      : v.FB
|  18  | --> pin route of INPUT4       : drdcen
|  19  | --> internal route of H1      : i0.FB
|  20  | --> pin route of INPUT6       : subwr
|  21  | --> pin route of INPUT1       : cacs
|  22  | --> pin route of C6           : ads
|  23  | --> pin route of INPUT5       : drack
|  24  | --> internal route of C5      : e.FB
|  25  | --> internal route of D6      : oeareg.Q

26 of 26 routes used (utilization = 100.0%)

---------------------
 Routes for block B
---------------------
+-----------------------------------------------+
|input -->           path              : signal |
+-----------------------------------------------+
|   0  | -->
|   1  | --> internal route of B9      : wc1.Q
|   2  | --> internal route of B3      : n2.FB
|   3  | --> pin route of A6           : jmpc0
|   4  | --> internal route of A3      : n0.FB
|   5  | --> internal route of B7      : wc2.Q
|   6  | --> pin route of B2           : blast
|   7  | --> internal route of C1      : nr.FB
|   8  | --> pin route of C4           : suba3
|   9  | --> pin route of INPUT7       : res
| 10 | -->
| 11 | --> internal route of A9      : n1.FB
| 12 | --> pin route of C2           : suba2
| 13 | --> internal route of C9      : cm.FB
| 14 | --> pin route of A10          : mrq
| 15 | --> internal route of C5      : e.FB
| 16 | --> internal route of B5      : wc3.Q
| 17 | --> internal route of C3      : v.FB
| 18 | --> pin route of INPUT4       : drdcen
| 19 | --> pin route of C0           : burstc
| 20 | --> internal route of B0      : ld.Q
| 21 | --> internal route of B11     : wc0.Q
| 22 | --> pin route of D8           : suba5
| 23 | --> pin route of C8           : suba4
| 24 | --> pin route of INPUT2       : jmpc1
| 25 | --> pin route of INPUT5       : drack

24 of 26 routes used (utilization = 92.3%)

---------------------
 Routes for block C
---------------------
+-----------------------------------------------+
|input -->           path           : signal |
+-----------------------------------------------+
|  0 | --> pin route of INPUT7       : res
|  1 | --> internal route of C7      : cv.FB
|  2 | --> internal route of B3      : n2.FB
|  3 | --> pin route of A6           : jmpc0
|  4 | --> pin route of INPUT3       : vsbcs
|  5 | --> pin route of F4           : vsbreq
|  6 | --> internal route of C9      : cm.FB
|  7 | --> internal route of C1      : nr.FB
|  8 | --> internal route of C3      : v.FB
|  9 | --> internal route of A3      : n0.FB
| 10 | --> internal route of A5      : np
| 11 | --> pin route of A4           : dramcs
| 12 | --> pin route of INPUT1       : cacs
| 13 | --> internal route of D7      : cyst
| 14 | --> pin route of C6           : ads
| 15 | --> internal route of D3      : ci.FB
| 16 | --> pin route of B4           : lock
| 17 | --> pin route of INPUT2       : jmpc1
| 18 | --> pin route of INPUT6       : subwr
| 19 | --> pin route of H10          : hold
| 20 | --> pin route of B8           : reade
| 21 | --> pin route of B10          : vsbread
| 22 | --> pin route of A8           : ethcs
| 23 | --> internal route of A9      : n1.FB
| 24 | --> internal route of C5      : e.FB
| 25 | --> pin route of INPUT4       : drdcen

26 of 26 routes used (utilization = 100.0%)

---------------------
 Routes for block D
```

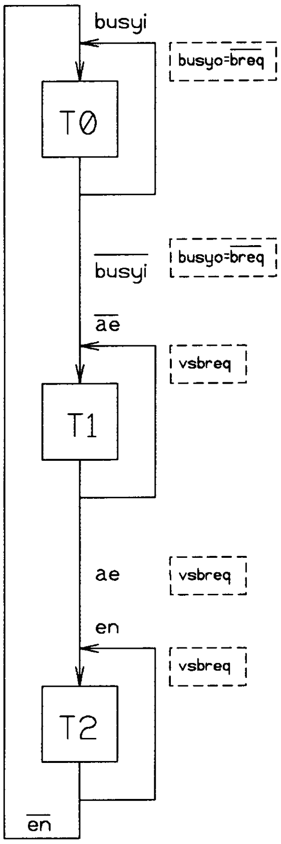
```
---------------------
+-----------------------------------------------+
|input -->            path                : signal |
+-----------------------------------------------+
|   0  | --> pin route of INPUT7           : res
|   1  | --> internal route of H5          : bc
|   2  | --> internal route of B3          : n2.FB
|   3  | --> pin route of INPUT5           : drack
|   4  | --> pin route of INPUT3           : vsbcs
|   5  | --> pin route of F4               : vsbreq
|   6  | --> internal route of C9          : cm.FB
|   7  | --> internal route of D7          : cyst
|   8  | --> pin route of B4               : lock
|   9  | --> internal route of A3          : n0.FB
|  10  | --> internal route of A5          : np
|  11  | --> pin route of A4               : dramcs
|  12  | --> pin route of INPUT1           : cacs
|  13  | --> pin route of A8               : ethcs
|  14  | --> internal route of D10         : oedreg
|  15  | --> internal route of D3          : ci.FB
|  16  | --> internal route of C1          : nr.FB
|  17  | --> internal route of C7          : cv.FB
|  18  | --> pin route of INPUT6           : subwr
|  19  | --> pin route of H10              : hold
|  20  | --> pin route of D4               : holda
|  21  | --> pin route of D6               : oeareg
|  22  | --> internal route of D0          : oevsba.Q
|  23  | --> internal route of A9          : n1.FB
|  24  | --> internal route of C5          : e.FB
|  25  | --> internal route of C3          : v.FB

26 of 26 routes used (utilization = 100.0%)

---------------------
 Routes for block E
---------------------
+-----------------------------------------------+
|input -->            path                : signal |
+-----------------------------------------------+
|   0  | -->
|   1  | --> internal route of C7          : cv.FB
|   2  | -->
|   3  | --> pin route of INPUT2           : jmpc1
|   4  | --> pin route of INPUT3           : vsbcs
|   5  | -->
|   6  | -->
|   7  | --> pin route of INPUT7           : res
|   8  | -->
|   9  | --> internal route of A3          : n0.FB
|  10  | --> pin route of B10              : vsbread
|  11  | --> pin route of A4               : dramcs
|  12  | -->
|  13  | --> internal route of C9          : cm.FB
|  14  | --> pin route of A10              : mrq
|  15  | --> internal route of B3          : n2.FB
|  16  | --> internal route of C1          : nr.FB
| 17 | --> internal route of C3      : v.FB
| 18 | --> pin route of INPUT4       : drdcen
| 19 | -->
| 20 | --> pin route of INPUT6       : subwr
| 21 | --> pin route of A6           : jmpc0
| 22 | --> pin route of C6           : ads
| 23 | --> internal route of A9      : n1.FB
| 24 | --> internal route of C5      : e.FB
| 25 | --> pin route of INPUT5       : drack

19 of 26 routes used (utilization = 73.1%)

---------------------
 Routes for block F
---------------------

+-----------------------------------------------+
|input -->             path             : signal |
+-----------------------------------------------+
|  0 | -->
|  1 | -->
|  2 | -->
|  3 | --> pin route of INPUT5       : drack
|  4 | -->
|  5 | --> pin route of D2           : mr
|  6 | --> pin route of B2           : blast
|  7 | -->
|  8 | -->
|  9 | --> internal route of A3      : n0.FB
| 10 | -->
| 11 | -->
| 12 | -->
| 13 | --> internal route of C9      : cm.FB
| 14 | --> pin route of A10          : mrq
| 15 | -->
| 16 | --> internal route of C1      : nr.FB
| 17 | --> internal route of C3      : v.FB
| 18 | --> pin route of INPUT4       : drdcen
| 19 | --> pin route of C0           : burstc
| 20 | -->
| 21 | --> pin route of F6           : dburst
| 22 | -->
| 23 | --> internal route of A9      : n1.FB
| 24 | --> internal route of C5      : e.FB
| 25 | --> internal route of B3      : n2.FB

14 of 26 routes used (utilization = 53.8%)

---------------------
 Routes for block G
---------------------

+-----------------------------------------------+
|input -->             path             : signal |
+-----------------------------------------------+
|  0 | -->
|  1 | -->
|  2 | -->
|   3  | -->
|   4  | -->
|   5  | -->
|   6  | -->
|   7  | --> pin route of INPUT7        : res
|   8  | --> internal route of C3       : v.FB
|   9  | --> internal route of A3       : n0.FB
|  10  | -->
|  11  | -->
|  12  | -->
|  13  | -->
|  14  | -->
|  15  | --> pin route of INPUT3        : vsbcs
|  16  | --> internal route of C1       : nr.FB
|  17  | --> internal route of C7       : cv.FB
|  18  | -->
|  19  | -->
|  20  | -->
|  21  | -->
|  22  | -->
|  23  | --> internal route of A9       : n1.FB
|  24  | --> pin route of INPUT2        : jmpc1
|  25  | --> internal route of B3       : n2.FB

9 of 26 routes used (utilization = 34.6%)

---------------------
 Routes for block H
---------------------
+-----------------------------------------------+
|input -->            path              : signal |
+-----------------------------------------------+
|   0  | -->
|   1  | -->
|   2  | -->
|   3  | -->
|   4  | -->
|   5  | -->
|   6  | -->
|   7  | --> pin route of INPUT7        : res
|   8  | -->
|   9  | --> internal route of H3       : i1.FB
|  10  | -->
|  11  | -->
|  12  | -->
|  13  | --> internal route of C9       : cm.FB
|  14  | -->
|  15  | --> internal route of D3       : ci.FB
|  16  | -->
|  17  | --> internal route of C7       : cv.FB
|  18  | -->
|  19  | --> internal route of H1       : i0.FB
|  20  | --> pin route of INPUT6        : subwr
|  21  | --> pin route of INPUT1        : cacs
|  22  | --> pin route of A8            : ethcs
|  23  | -->
| 24 | -->
| 25 | -->

9 of 26 routes used (utilization = 34.6%)

CHIP route summary: 153 of 208 routes used (utilization = 73.6%)
---------------------------------------------------


Routing Table:

BLOCK        SIGNAL
........        clk
ABCDE.GH        res
..CD....        vsbreq
.B..EF..        mrq
..C.E...        vsbread
ABC.E.G.        jmpc1
ABC.E...        jmpc0
.B......        suba5
.B......        suba4
.B......        suba3
.B......        suba2
.....F..        dburst
ABC.EF..        drdcen
AB.DEF..        drack
A....F..        mr
A.CDE...        dramcs
A.CDE.G.        vsbcs
..CD...H        ethcs
A.CD...H        cacs
A.CDE..H        subwr
AB...F..        burstc
..C.....        reade
A.C.E...        ads
AB...F..        blast
..CD....        hold
..CD....        lock
...D....        holda
...D....        oeareg
...D....        oedreg
A.CD....        np
..CD....        cyst
...D....        bc
ABCDEFG.        v.FB
A.CDE.GH        cv.FB
ABCDEFG.        nr.FB
A.......        oeareg.Q
A.CD...H        ci.FB
ABCDEF.H        cm.FB
ABCDEF..        e.FB
...D....        oevsba.Q
ABCDEFG.        n2.FB
ABCDEFG.        n1.FB
ABCDEFG.        n0.FB
.B......        wc0.Q
```

```
.B......       wc1.Q
.B......       wc2.Q
.B......       wc3.Q
.B......       ld.Q
A......H       i1.FB
A......H       i0.FB

Current partitioning requires 153 routes.
----------------------------------------------------
$DEVICE MACH220A fit cc0.tt3
$PINS 50 clk:15 res:54 vsbreq:46 mrq:7 vsbread:9 jmpc1:17 jmpc0:5 suba5:29
suba4:25 suba3:23 suba2:22 drdcen:49 drack:50 mr:32 dramcs:4 vsbcs:20
ethcs:6 cacs:16 subwr:51 burstc:21 reade:10 ads:24 blast:13 hold:62 lock:12
ld+:14 oevsba-:33 oedreg+:28 crdcen+:2 holda+:31 dle+:36 oeareg+:30 dale+:3
7
oevsbd-:38 cack+:3 dcs+:39 mack+:40 burstdm+:41 erdy+:11 ebrdy+:48 waitd+:4
7
dburst+:45 vsbaclk+:55 ena+:56 port+:67 start+:57 ca+:66 dwr+:58 mast+:59
en+:60
$NODES 18 n0:70 nr:81 np:71 wc3:77 v:82 n2:76 n1:73 wc2:78 e:83
wc1:79 wc0:80 ci:88 cv:84 cm:85 cyst:90 i0:111 i1:112 bc:113
----------------------------------------------------
```

VSB Control Overview	KAN	Page 1
	921220	VSB



VSB Control Arbitration. Synchronous Part	KAN	Page 2
	921220	VSB



VSB Control Arbitration. Asynchronous Part	KAN	Page 3
	921220	VSB

en, busy

00	01	11	10	bgout
Ⓐ	B	-	C	0
A	Ⓑ	D	-	0
-	B	Ⓓ	C	0
A	-	E	Ⓒ	1
-	B	Ⓔ	F	0
A	-	E	Ⓕ	0

en, busy

00	01	11	10	bgout
Ⓐ	Ⓑ	Ⓓ	C	0
A	-	E	Ⓒ	1
A	B	Ⓔ	Ⓕ	0

en, busy

ae,y0	00	01	11	10
00	00	00	00	01
01	00	00	11	01
11	01	01	11	11

	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	1	1
10				

ae

	00	01	11	10
00	0	0	0	1
01	0	0	1	1
11	1	1	1	1
10				

y0

	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	0	0	0	0
10	0	0	0	0

bgout

ae = en busy y0 + en ae

y0 = ae + y0 en + en $\overline{\text{busy}}$

bgout = $\overline{\text{ae}}$ y0 en

VSB Control Handshake in address phase	KAN 921222	Page 4 VSB

ena,pas	00	01	11	10	pasl
	Ⓐ	C	-	B	0
	A	-	D	Ⓑ	0
	-	Ⓒ	D	-	0
	-	-	Ⓓ	E	1
	A	-	F	Ⓔ	0
	-	C	Ⓕ	-	0

ena,pas	00	01	11	10
00	Ⓐ	Ⓒ	D	Ⓑ
01	-	-	Ⓓ	E
11	A	-	F	Ⓔ
10	-	C	Ⓕ	-

y2,y1 \ ena,pas	00	01	11	10
00	00	00	01	00
01	00	00	01	11
11	00	00	10	11
10	00	00	10	- -

	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	0	0	1	1
10	0	0	1	

y2

	00	01	11	10
00	0	0	1	0
01	0	0	1	1
11	0	0	0	1
10	0	0	0	

y1

	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	0	0
10	0	0	0	0

pasl

	00	01	11	10
00	1	1	1	1
01	1	1	0	0
11	1	1	0	0
10	1	1	0	0

levsba

$$y2 = y2\ ena + y1\ ena\ \overline{pas}$$

$$y1 = \overline{y2}\ ena\ pas + y1\ ena\ \overline{pas} + \overline{y2}\ y1\ ena$$

$$pasl = \overline{y2}\ y1\ ena\ pas$$

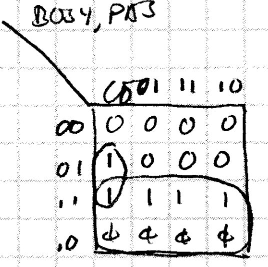
$$levsba = \overline{y2}\ \overline{y1} + \overline{ena}$$

Initials KBN	Page
Date	Project

ENABLES of END

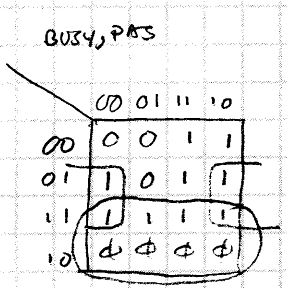
BUSY, PAS	00	01	11	10
00	A	A	B	B
01	C	A	B	B
11	C	C	C	C

END1 ENDO \ BUSY, PAS	00	01	11	10
00	00	00	01	01
01	11	00	01	01
11	11	11	11	11



BUSY, PAS	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	1	1
10	Φ	Φ	Φ	Φ

END1



BUSY, PAS	00	01	11	10
00	0	0	1	1
01	1	0	1	1
11	1	1	1	1
10	Φ	Φ	Φ	Φ

ENDO

$\text{END1} = \text{END1} + \overline{\text{BUSY}} \cdot \overline{\text{PAS}} \cdot \text{ENDO}$; $\text{ENDO} = \text{END1} + \text{ENDO} \cdot \overline{\text{PAS}} + \text{BUSY}$

VSB control Handshake in address phase	KAN	Page 5
	921222	VSB

pasl,y4	00	01	11	10
	Ⓐ	–	–	B
	–	–	C	Ⓑ
	–	–	Ⓒ	D
	A	–	–	Ⓓ

alt skal
enables af ena

y5 \ pasl,y4	00	01	11	10
0	Ⓐ	–	C	Ⓑ
1	A	–	Ⓒ	Ⓓ

y5 \ pasl,y4	00	01	11	10
0	0	–	1	0
1	0	–	1	1

y5 = pasl y4 + y5 pasl

y5 \ pasl,y4	00	01	11	10
0	0	–	1	1
1	0	–	0	0

y3 = $\overline{\text{y5}}$ pasl

y5 \ pasl,y4	00	01	11	10
0	0	–	0	0
1	0	–	0	1

acx = y5 pasl $\overline{\text{y4}}$

asack = y4 pasl + asack pas

VSB control Handshake in address phase	KAN	Page 6
	921222	VSB

y6 \ acx/acnul	00	01	11	10
	Ⓐ	Ⓑ	C	Ⓓ
	A	B	Ⓒ	Ⓔ

y6 \ acx/acnul	00	01	11	10
0	0	0	1	0
1	0	0	1	1

$$y6 = acx\ acnul + y6\ acx$$

y6 \ acx/acnul	00	01	11	10
0	0	0	1	1
1	0	0	0	0

$$ac = \overline{y6}\ acx$$

enables af ena

VSB Control Handshake in data phase	KAN 901224	Page 7 VSB

dr,ds	00	01	11	10	ack
	A	B	–	–	0
	–	B	C	–	0
	–	G	C	D	1
	A	–	E	D	0
	–	B	E	–	0
	A	G	–	–	1

dr,ds	00	01	11	10	ack
	A	B	C		0
	A	B	E	D	0
	A	G	C	D	1

y8,y7 \ dr,ds	00	01	11	10
00	00	00	10	01
01	00	00	01	01
11	00	00	00	00
10	00	10	10	00

y8,y7 \ dr,ds	00	01	11	10
00	0	0	1	0
01	0	0	0	0
11	0	0	0	0
10	0	1	1	0

y8

y8,y7 \ dr,ds	00	01	11	10
00	0	0	0	1
01	0	0	1	1
11	0	0	0	0
10	0	0	0	0

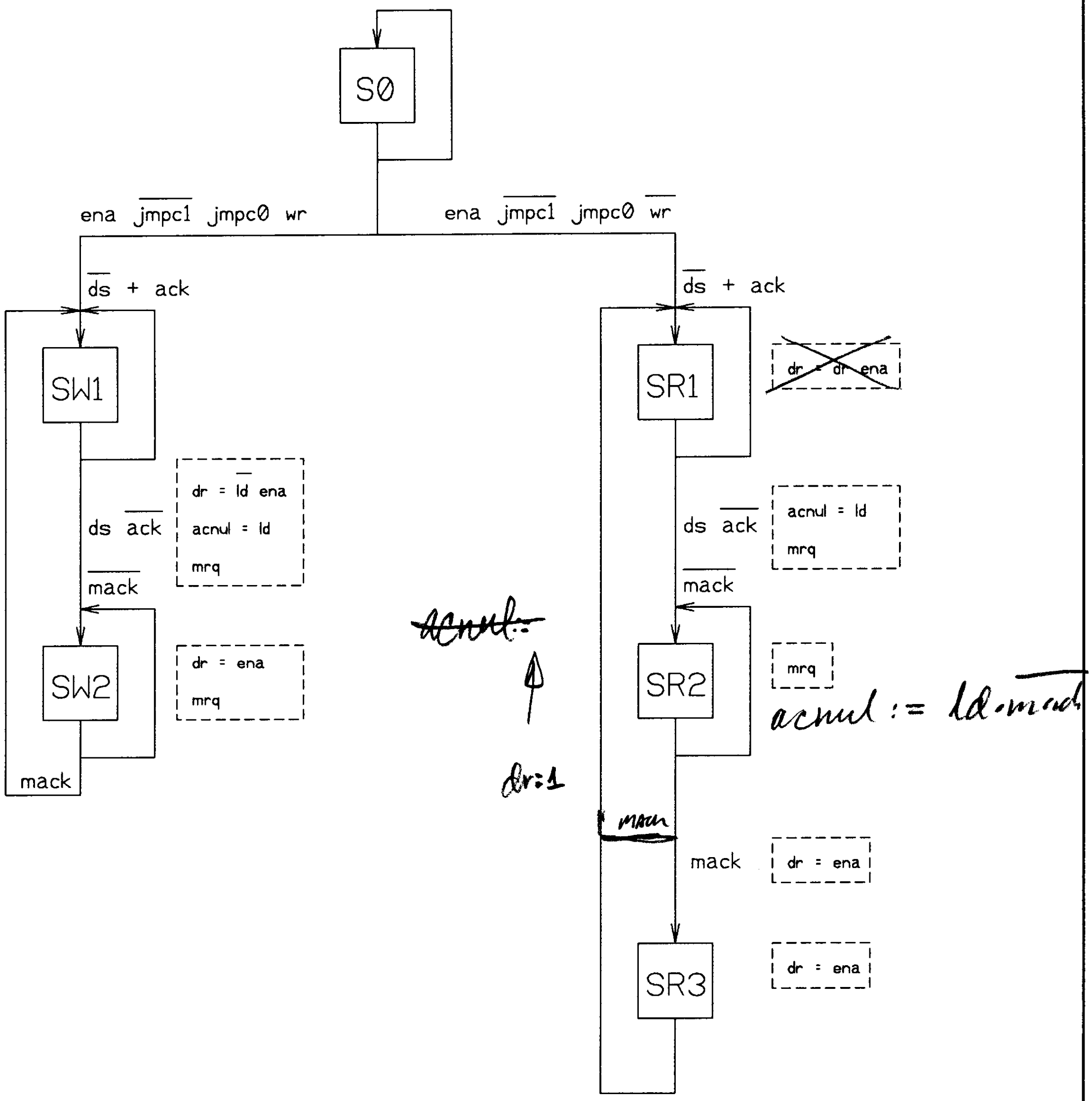
y7

$y8 = dr\ ds\ \overline{y7} + ds\ y8\ \overline{y7}$

$y7 = dr\ \overline{y8}\ y7 + dr\ \overline{ds}\ \overline{y7}\ \overline{y8}$ KAN 951026

$ack = y8$

VSB Control Data phase	KAN	Page 8
	921224	VSB



```
----------------------------------------------------

User pre-assignments:

---------------------------------------------------
Results of fitting vsb.tt2 into MACH210.
---------------------------------------------------

---------- Dedicated Inputs --------
PIN:10  --> wri
PIN:11  --> pasi
PIN:32  --> start
PIN:33  --> res

---------- Clock Inputs ------------
PIN:13  --> clk
PIN:35  --> dsi


-------------------------------------
       Resources of block A
-------------------------------------
Inputs routed to block:  res ena ld dsi wri mack acko y7 y8 dr jmpc1.FB
jmpc0.FB mrq.FB n2.FB n3.FB

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  A0   (PIN :2   ) -->
     PIN OUTPUT          : mrq.REG (pts=4)
     PRODUCT TERMS       : DOWN
  A1   (NODE:45  ) -->
     INTERNAL FEEDBACK : dr.REG (pts=6)
     PRODUCT TERMS       : LOCAL
  A2   (PIN :3   ) -->
     PIN INPUT           : acko
     PIN OUTPUT          : acko (pts=1)
     PRODUCT TERMS       : UP2
  A3   (NODE:46  ) -->
     INTERNAL FEEDBACK : n2.REG (pts=3)
     PRODUCT TERMS       : LOCAL
  A4   (PIN :4   ) -->
     PIN INPUT           : ena
     PRODUCT TERMS       : UP2
  A5   (NODE:47  ) -->
     INTERNAL FEEDBACK : n3.REG (pts=3)
     PRODUCT TERMS       : LOCAL
  A6   (PIN :5   ) -->
     PIN INPUT           : en
     PRODUCT TERMS       : NOT USED
  A7   (NODE:48 ) -->
     INTERNAL FEEDBACK : acnul.REG (pts=2)
     PRODUCT TERMS     : LOCAL
  A8   (PIN :6   ) -->
     PIN INPUT         : busyi
     PRODUCT TERMS     : NOT USED
  A9   (NODE:49 ) -->
     INTERNAL FEEDBACK : y8 (pts=2)
     PRODUCT TERMS     : LOCAL
  A10  (PIN :7   ) -->
     PIN INPUT         : breq
     PRODUCT TERMS     : NOT USED
  A11  (NODE:50 ) -->
     INTERNAL FEEDBACK : y7 (pts=2)
     PRODUCT TERMS     : LOCAL
  A12  (PIN :8   ) -->
     PIN INPUT         : read
     PRODUCT TERMS     : NOT USED
  A13  (NODE:51 ) -->
     PRODUCT TERMS     : NOT USED
  A14  (PIN :9   ) -->
     PIN INPUT         : mast
     PRODUCT TERMS     : NOT USED
  A15  (NODE:52 ) -->
     PRODUCT TERMS     : NOT USED

------------------------------------
       Block A resource summary:

              USED    FREE    UTILIZATION
  product terms 23      41       35.9%
  macro cells    9       7       56.3%
  pins           8       0      100.0%
------------------------------------



------------------------------------
       Resources of block B
------------------------------------
Inputs routed to block:  pasi y4 asacko pasl y5 y1s y2s busyd aci asacki
acki mast start y9

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  B0   (PIN :21 ) -->
     PIN OUTPUT        : jmpc0.REG (pts=4)
     PRODUCT TERMS     : LOCAL
  B1   (NODE:53 ) -->
```

```
     INTERNAL FEEDBACK : y9 (pts=2)
     PRODUCT TERMS     : LOCAL
  B2   (PIN :20 ) -->
     PIN OUTPUT        : jmpcl.REG (pts=3)
     PRODUCT TERMS     : LOCAL
  B3   (NODE:54 ) -->
     INTERNAL FEEDBACK : y5 (pts=2)
     PRODUCT TERMS     : LOCAL
  B4   (PIN :19 ) -->
     PIN INPUT         : dso
     PIN OUTPUT        : dso (pts=1)
     PRODUCT TERMS     : LOCAL
  B5   (NODE:55 ) -->
     INTERNAL FEEDBACK : acx (pts=1)
     PRODUCT TERMS     : LOCAL
  B6   (PIN :18 ) -->
     PIN INPUT         : asacko
     PIN OUTPUT        : asacko (pts=2)
     PRODUCT TERMS     : LOCAL
  B7   (NODE:56 ) -->
     PRODUCT TERMS     : NOT USED
  B8   (PIN :17 ) -->
     PIN OUTPUT        : y3 (pts=2)
     PRODUCT TERMS     : LOCAL
  B9   (NODE:57 ) -->
     PRODUCT TERMS     : NOT USED
  B10  (PIN :16 ) -->
     PIN INPUT         : vsbaoe
     PIN OUTPUT        : vsbaoe (pts=1)
     PRODUCT TERMS     : LOCAL
  B11  (NODE:58 ) -->
     PRODUCT TERMS     : NOT USED
  B12  (PIN :15 ) -->
     PIN INPUT         : acki
     PRODUCT TERMS     : NOT USED
  B13  (NODE:59 ) -->
     PRODUCT TERMS     : NOT USED
  B14  (PIN :14 ) -->
     PIN INPUT         : asacki
     PRODUCT TERMS     : NOT USED
  B15  (NODE:60 ) -->
     PRODUCT TERMS     : NOT USED

 ------------------------------------
       Block B resource summary:

                  USED     FREE     UTILIZATION
  product terms    18       46        28.1%
  macro cells       9        7        56.3%
  pins              8        0       100.0%
 ------------------------------------



 ------------------------------------
       Resources of block C
```

```
------------------------------------
Inputs routed to block:  ena pasi y1 y2 busys dsi wri dr start read dso
vsbaoe vsbdoe

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  C0    (PIN :24 ) -->
     PIN INPUT            : vsbdoe
     PIN OUTPUT           : vsbdoe (pts=3)
     PRODUCT TERMS        : LOCAL
  C1    (NODE:61 ) -->
     INTERNAL FEEDBACK : y1 (pts=3)
     PRODUCT TERMS        : LOCAL
  C2    (PIN :25 ) -->
     PIN OUTPUT           : levbsd (pts=2)
     PRODUCT TERMS        : LOCAL
  C3    (NODE:62 ) -->
     INTERNAL FEEDBACK : y2 (pts=2)
     PRODUCT TERMS        : LOCAL
  C4    (PIN :26 ) -->
     PIN OUTPUT           : levsba (pts=2)
     PRODUCT TERMS        : LOCAL
  C5    (NODE:63 ) -->
     INTERNAL FEEDBACK : pasl (pts=1)
     PRODUCT TERMS        : LOCAL
  C6    (PIN :27 ) -->
     PIN INPUT            : aci
     PRODUCT TERMS        : NOT USED
  C7    (NODE:64 ) -->
     INTERNAL FEEDBACK : y2s.REG (pts=1)
     PRODUCT TERMS        : LOCAL
  C8    (PIN :28 ) -->
     PIN INPUT            : y4
     PRODUCT TERMS        : NOT USED
  C9    (NODE:65 ) -->
     INTERNAL FEEDBACK : y1s.REG (pts=1)
     PRODUCT TERMS        : LOCAL
  C10   (PIN :29 ) -->
     PIN INPUT            : mack
     PRODUCT TERMS        : NOT USED
  C11   (NODE:66 ) -->
     INTERNAL FEEDBACK : busyd.REG (pts=1)
     PRODUCT TERMS        : LOCAL
  C12   (PIN :30 ) -->
     PIN INPUT            : ld
     PRODUCT TERMS        : NOT USED
  C13   (NODE:67 ) -->
     PRODUCT TERMS        : NOT USED
  C14   (PIN :31 ) -->
```

```
     PRODUCT TERMS      : NOT USED
  C15  (NODE:68 ) -->
     PRODUCT TERMS      : NOT USED

------------------------------------
       Block C resource summary:

                 USED    FREE    UTILIZATION
  product terms  16      48       25.0%
  macro cells     9       7       56.3%
  pins            7       1       87.5%
------------------------------------



------------------------------------
       Resources of block D
------------------------------------
Inputs routed to block:  res breq busyi en y0 ae acx y6 acnul n1.FB n0.FB

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  D0   (PIN :43 ) -->
     PIN OUTPUT         : vsbreq.REG (pts=2)
     PRODUCT TERMS      : LOCAL
  D1   (NODE:69 ) -->
     INTERNAL FEEDBACK  : n1.REG (pts=2)
     PRODUCT TERMS      : LOCAL
  D2   (PIN :42 ) -->
     PIN OUTPUT         : busyo (pts=1)
     PRODUCT TERMS      : LOCAL
  D3   (NODE:70 ) -->
     INTERNAL FEEDBACK  : n0.REG (pts=3)
     PRODUCT TERMS      : LOCAL
  D4   (PIN :41 ) -->
     PIN OUTPUT         : bgout (pts=1)
     PRODUCT TERMS      : LOCAL
  D5   (NODE:71 ) -->
     INTERNAL FEEDBACK  : ae (pts=2)
     PRODUCT TERMS      : LOCAL
  D6   (PIN :40 ) -->
     PIN OUTPUT         : aco (pts=1)
     PRODUCT TERMS      : LOCAL
  D7   (NODE:72 ) -->
     INTERNAL FEEDBACK  : y0 (pts=3)
     PRODUCT TERMS      : LOCAL
  D8   (PIN :39 ) -->
     PRODUCT TERMS      : NOT USED
  D9   (NODE:73 ) -->
```

```
     INTERNAL FEEDBACK : y6 (pts=2)
     PRODUCT TERMS     : LOCAL
  D10  (PIN :38 ) -->
     PRODUCT TERMS     : NOT USED
  D11  (NODE:74 ) -->
     INTERNAL FEEDBACK : busys.REG (pts=1)
     PRODUCT TERMS     : LOCAL
  D12  (PIN :37 ) -->
     PRODUCT TERMS     : NOT USED
  D13  (NODE:75 ) -->
     PRODUCT TERMS     : NOT USED
  D14  (PIN :36 ) -->
     PRODUCT TERMS     : NOT USED
  D15  (NODE:76 ) -->
     PRODUCT TERMS     : NOT USED

------------------------------------
       Block D resource summary:

                  USED     FREE    UTILIZATION
  product terms   18        46       28.1%
  macro cells     10         6       62.5%
  pins             4         4       50.0%
------------------------------------



------------------------------------
       CHIP resource summary:

                  USED     FREE    UTILIZATION
  product terms   75       181       29.3%
  macro cells     37        27       57.8%
  pins            33         5       86.8%

------------------------------------

----------------------------------------------------
Routing results.
----------------------------------------------------

---------------------
 Routes for block A
---------------------
+-----------------------------------------------+
|input -->             path               : signal |
+-----------------------------------------------+
|   0  | -->
|   1  | -->
|   2  | --> pin route of A4            : ena
|   3  | --> internal route of A9       : y8
|   4  | -->
|   5  | --> internal route of A11      : y7
|   6  | -->
|   7  | -->
|   8  | --> pin route of INPUT5        : dsi
|  9  | -->
| 10  | --> pin route of INPUT0       : wri
| 11  | --> pin route of INPUT4       : res
| 12  | --> pin route of A2           : acko
| 13  | --> internal route of A0      : mrq.FB
| 14  | --> internal route of A1      : dr
| 15  | -->
| 16  | --> internal route of A3      : n2.FB
| 17  | --> pin route of C10          : mack
| 18  | --> internal route of A5      : n3.FB
| 19  | --> internal route of B0      : jmpc0.FB
| 20  | --> pin route of C12          : ld
| 21  | --> internal route of B2      : jmpc1.FB

15 of 22 routes used (utilization = 68.2%)

---------------------
 Routes for block B
---------------------

+----------------------------------------------+
|input -->           path          : signal |
+----------------------------------------------+
|  0  | --> internal route of C11     : busyd
|  1  | -->
|  2  | -->
|  3  | -->
|  4  | -->
|  5  | --> internal route of B1      : y9
|  6  | --> pin route of INPUT3       : start
|  7  | -->
|  8  | -->
|  9  | --> internal route of C9      : y1s
| 10  | --> pin route of C8           : y4
| 11  | --> internal route of B3      : y5
| 12  | --> pin route of B12          : acki
| 13  | --> pin route of B14          : asacki
| 14  | -->
| 15  | --> pin route of C6           : aci
| 16  | --> pin route of INPUT1       : pasi
| 17  | -->
| 18  | --> pin route of A14          : mast
| 19  | --> internal route of C5      : pasl
| 20  | --> pin route of B6           : asacko
| 21  | --> internal route of C7      : y2s

14 of 22 routes used (utilization = 63.6%)

---------------------
 Routes for block C
---------------------

+----------------------------------------------+
|input -->           path          : signal |
+----------------------------------------------+
|  0  | -->
|  1  | -->
|  2  | --> pin route of A4           : ena
```

```
 |  3  | -->
 |  4  | -->
 |  5  | -->
 |  6  | --> pin route of INPUT3        : start
 |  7  | -->
 |  8  | -->
 |  9  | -->
 | 10  | --> pin route of INPUT0        : wri
 | 11  | --> pin route of B10           : vsbaoe
 | 12  | --> pin route of C0            : vsbdoe
 | 13  | --> pin route of INPUT5        : dsi
 | 14  | --> internal route of A1       : dr
 | 15  | --> internal route of C1       : y1
 | 16  | --> pin route of INPUT1        : pasi
 | 17  | --> internal route of C3       : y2
 | 18  | --> internal route of D11      : busys
 | 19  | --> pin route of B4            : dso
 | 20  | --> pin route of A12           : read
 | 21  | -->

13 of 22 routes used (utilization = 59.1%)

---------------------
 Routes for block D
---------------------
+------------------------------------------+
|input -->          path           : signal |
+------------------------------------------+
 |  0  | -->
 |  1  | -->
 |  2  | -->
 |  3  | -->
 |  4  | -->
 |  5  | --> pin route of A10           : breq
 |  6  | -->
 |  7  | -->
 |  8  | --> internal route of A7       : acnul
 |  9  | -->
 | 10  | -->
 | 11  | --> pin route of INPUT4        : res
 | 12  | -->
 | 13  | --> internal route of B5       : acx
 | 14  | --> pin route of A6            : en
 | 15  | --> pin route of A8            : busyi
 | 16  | --> internal route of D7       : y0
 | 17  | --> internal route of D3       : n0.FB
 | 18  | -->
 | 19  | --> internal route of D5       : ae
 | 20  | --> internal route of D9       : y6
 | 21  | --> internal route of D1       : n1.FB

11 of 22 routes used (utilization = 50.0%)

CHIP route summary: 53 of 88 routes used (utilization = 60.2%)
---------------------------------------------------
```

```
Routing Table:

BLOCK       SIGNAL
....       clk
A..D       res
...D       breq
...D       busyi
...D       en
...D       y0
...D       ae
A.C.       ena
.BC.       pasi
A...       ld
.B..       y4
.B..       asacko
.B..       pasl
...D       acx
..C.       y1
..C.       y2
.B..       y5
...D       y6
.B..       y1s
.B..       y2s
..C.       busys
.B..       busyd
A.C.       dsi
A.C.       wri
A...       mack
A...       acko
A...       y7
A...       y8
A.C.       dr
...D       acnul
.B..       aci
.B..       asacki
.B..       acki
.B..       mast
.BC.       start
..C.       read
..C.       dso
..C.       vsbaoe
.B..       y9
..C.       vsbdoe
...D       n1.FB
...D       n0.FB
A...       jmpc1.FB
A...       jmpc0.FB
A...       mrq.FB
A...       n2.FB
A...       n3.FB

Current partitioning requires 53 routes.
----------------------------------------------------
$DEVICE MACH210A fit vsb.tt3
$PINS 33 clk:13 res:33 breq:7 busyi:6 en:5 ena:4 pasi:11 ld:30 y4:28
```

```
dsi:35 wri:10 mack:29 aci:27 asacki:14 acki:15 mast:9 start:32 read:8
mrq+:2 jmpc0+:21 jmpc1+:20 vsbdoe+:24 dso+:19 asacko+:18 y3+:17 vsbreq+:43
busyo+:42 levbsd+:25 levsba+:26 bgout+:41 vsbaoe+:16 aco+:40 acko+:3
$NODES 21 dr:45 n2:46 n3:47 acnul:48 n1:69 n0:70 y1:61 y2:62 pasl:63
ae:71 y8:49 y0:72 y7:50 y9:53 y5:54 y6:73 acx:55 y2s:64 y1s:65
busyd:66 busys:74
----------------------------------------------------
```

```
   module rfcnt
   title 'refresh counter 921208 kan'

   rfcnt device 'P16R8';

   "inputs
   clk             pin 1; "clock input 30 MHz
   rfclk           pin 2; "Refresh clock input
   res             pin 9; "reset for testing only

   "outputs

   oe              pin 11; "output enable
   rx              pin 13 istype 'reg_d, invert'; "rfclk synchronized
   ry              pin 14 istype 'reg_d, invert';
   cnt             pin 15 istype 'reg_d, invert'; "count counter
   cc0             pin 16 istype 'reg_d, invert'; "least significant counter
bit
   cc1             pin 17 istype 'reg_d, invert';
   cc2             pin 18 istype 'reg_d, invert';
   cc3             pin 19 istype 'reg_d, invert'; "most significant counter b
it

   c,x,z    =      .C., .X., .Z.;

   equations;

   rx.c  = clk;
   ry.c  = clk;
   cnt.c = clk;
   cc0.c = clk;
   cc1.c = clk;
   cc2.c = clk;
   cc3.c = clk;

   rx.d = rfclk;
   ry.d = rx.q;
   cnt.d = !ry.q & rx.q;

   declarations;

   s0  = ^b0000;
   s1  = ^b0001;
   s2  = ^b0010;
   s3  = ^b0011;
   s4  = ^b0100;
   s5  = ^b0101;
   s6  = ^b0110;
   s7  = ^b0111;
   s8  = ^b1000;
   s9  = ^b1001;
   s10 = ^b1010;
   s11 = ^b1011;
   s12 = ^b1100;
   s13 = ^b1101;
   s14 = ^b1110;
   s15 = ^b1111;
```

```
counter = [cc3, cc2, cc1, cc0];
inp     = [cnt.q, res];

state_diagram counter;

state s0 : case ( inp == [0, 0] ) : s0;
                ( inp == [1, 0] ) : s1;
                ( inp == [x, 1] ) : s0;
           endcase;

state s1 : case ( inp == [0, 0] ) : s1;
                ( inp == [1, 0] ) : s2;
                ( inp == [x, 1] ) : s0;
           endcase;

state s2 : case ( inp == [0, 0] ) : s2;
                ( inp == [1, 0] ) : s3;
                ( inp == [x, 1] ) : s0;
           endcase;

state s3 : case ( inp == [0, 0] ) : s3;
                ( inp == [1, 0] ) : s4;
                ( inp == [x, 1] ) : s0;
           endcase;

state s4 : case ( inp == [0, 0] ) : s4;
                ( inp == [1, 0] ) : s5;
                ( inp == [x, 1] ) : s0;
           endcase;

state s5 : case ( inp == [0, 0] ) : s5;
                ( inp == [1, 0] ) : s6;
                ( inp == [x, 1] ) : s0;
           endcase;

state s6 : case ( inp == [0, 0] ) : s6;
                ( inp == [1, 0] ) : s7;
                ( inp == [x, 1] ) : s0;
           endcase;

state s7 : case ( inp == [0, 0] ) : s7;
                ( inp == [1, 0] ) : s8;
                ( inp == [x, 1] ) : s0;
           endcase;

state s8 : case ( inp == [0, 0] ) : s8;
                ( inp == [1, 0] ) : s9;
                ( inp == [x, 1] ) : s0;
           endcase;

state s9 : case ( inp == [0, 0] ) : s9;
                ( inp == [1, 0] ) : s10;
                ( inp == [x, 1] ) : s0;
           endcase;

state s10: case ( inp == [0, 0] ) : s10;
```

```
                ( inp == [1, 0] ) : s11;
                ( inp == [x, 1] ) : s0;
           endcase;

state s11: case ( inp == [0, 0] ) : s11;
                ( inp == [1, 0] ) : s12;
                ( inp == [x, 1] ) : s0;
           endcase;

state s12: case ( inp == [0, 0] ) : s12;
                ( inp == [1, 0] ) : s13;
                ( inp == [x, 1] ) : s0;
           endcase;

state s13: case ( inp == [0, 0] ) : s13;
                ( inp == [1, 0] ) : s14;
                ( inp == [x, 1] ) : s0;
           endcase;

state s14: case ( inp == [0, 0] ) : s14;
                ( inp == [1, 0] ) : s15;
                ( inp == [x, 1] ) : s0;
           endcase;

state s15: case ( inp == [0, 0] ) : s15
                ( inp == [1, 0] ) : s0;
                ( inp == [x, 1] ) : s0;
           endcase;


test_vectors
([clk, rfclk, res] -> [!rx, !ry, !cnt, counter]);

[ c, 0, 1 ] -> [ 0, x, x, s0  ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s0  ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s0  ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s0  ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s1  ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s1  ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s1  ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s1  ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s1  ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s2  ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s2  ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s2  ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s2  ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s2  ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s3  ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s3  ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s3  ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s3  ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s3  ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s4  ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s4  ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s4  ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s4  ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s4  ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s5  ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s5  ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s5  ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s5  ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s5  ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s6  ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s6  ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s6  ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s6  ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s6  ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s7  ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s7  ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s7  ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s7  ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s7  ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s8  ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s8  ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s8  ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s8  ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s8  ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s9  ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s9  ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s9  ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s9  ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s9  ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s10 ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s10 ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s10 ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s10 ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s10 ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s11 ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s11 ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s11 ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s11 ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s11 ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s12 ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s12 ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s12 ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s12 ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s12 ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s13 ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s13 ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s13 ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s13 ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s13 ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s14 ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s14 ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s14 ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s14 ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s14 ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s15 ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s15 ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s15 ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s15 ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s15 ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s0  ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s0  ] ;
[ c, 0, 0 ] -> [ 0, 0, 0, s0  ] ;

[ c, 1, 0 ] -> [ 1, 0, 0, s0  ] ;
[ c, 1, 0 ] -> [ 1, 1, 1, s0  ] ;
[ c, 1, 0 ] -> [ 1, 1, 0, s1  ] ;
[ c, 0, 0 ] -> [ 0, 1, 0, s1  ] ;
[ c, 0, 1 ] -> [ 0, 0, 0, s0  ] ;

end rfcnt
```

pal er delt i 2
pard : data
parp : program

```
"parity checker with status read out             kan 921209

module par1;

declarations;

clk                pin;               "clock input.

read               pin;               "output status; clear error

cs                 pin;               "chip select.

"checka             pin;                check input first level

"data is checked during all accesses to memory:
"parity must be valid during write cycles, when drack = 1;
"parity must be valid during read cycles, when drdcen = 1;

"during accesses to vsb bus parity must be valid when:
"mast & !jmpc1 & jmpc0;
"if mast & jmpc1 & jmpc0 = 1, no vsb module answers.

"checka is replaced by drack # drdcen # mast & !jmpc1 & jmpc0

drack              pin;               "acknowledge from DRAM write cycles
drdcen             pin;               "acknowledge from DRAM read cycles

mast               pin;               "BAIO master on vsb. from PAL cc0.
jmpc1              pin;               "jump code from PAL vsb
jmpc0              pin;               "jump code from PAL vsb


"the four status pins stata(3:0) are connected to the following
"output enable signals:

"oee    : output enable even word from DRAM.
"oeo    : output enable odd word from DRAM.
"oedreg : output enable DREG. data from CPU.
"oevsbd : output enable VSBD. data from vsb bus.

"if a parity error occurs and no output enable is activated,
"then the ethernet controller is responsible for driving SUBD.

stata3             pin;               "cycle status input pin bit 3
stata2             pin;               "cycle status input pin bit 2
stata1             pin;               "cycle status input pin bit 1
stata0             pin;               "cycle status input pin bit 0

bea3               pin;               "byte enable input pin da(31:24)
bea2               pin;               "byte enable input pin da(23:16)
bea1               pin;               "byte enable input pin da(15: 8)
bea0               pin;               "byte enable input pin da( 7: 0)

pa3     pin;                              "parity bit input pin da(31:24)
pa2     pin;                              "parity bit input pin da(23:16)
```

```
pa1  pin 48 istype'reg_d,buffer';   "parity bit i/o pin da(15: 8)
pa0  pin 67 istype'reg_d,buffer';   "parity bit i/o pin da( 7: 0)

da31            pin;            "data bit input pin
da30            pin;            "data bit input pin
da29            pin;            "data bit input pin
da28            pin;            "data bit input pin
da27            pin;            "data bit input pin
da26            pin;            "data bit input pin
da25            pin;            "data bit input pin
da24            pin;            "data bit input pin

da23            pin;            "data bit input pin
da22            pin;            "data bit input pin
da21            pin;            "data bit input pin
da20            pin;            "data bit input pin
da19            pin;            "data bit input pin
da18            pin;            "data bit input pin
da17            pin;            "data bit input pin
da16            pin;            "data bit input pin

da15    pin 24  istype'reg_d,buffer'; "data bit i/o pin
da14    pin 31  istype'reg_d,buffer'; "data bit i/o pin
da13    pin 30  istype'reg_d,buffer'; "data bit i/o pin
da12    pin 29  istype'reg_d,buffer'; "data bit i/o pin
da11    pin 13  istype'reg_d,buffer'; "data bit i/o pin
da10    pin 12  istype'reg_d,buffer'; "data bit i/o pin
da9     pin 11  istype'reg_d,buffer'; "data bit i/o pin
da8     pin 22  istype'reg_d,buffer'; "data bit i/o pin
da7     pin 23  istype'reg_d,buffer'; "data bit i/o pin
da6     pin 33  istype'reg_d,buffer'; "data bit i/o pin
da5     pin 32  istype'reg_d,buffer'; "data bit i/o pin
da4     pin 36  istype'reg_d,buffer'; "data bit i/o pin
da3     pin 21  istype'reg_d,buffer'; "data bit i/o pin
da2     pin 6   istype'reg_d,buffer'; "data bit i/o pin
da1     pin 5   istype'reg_d,buffer'; "data bit i/o pin
da0     pin 2   istype'reg_d,buffer'; "data bit i/o pin

pchki   pin                     ; "parity error detected by
                                  "ethernet controller.
pchk    node istype 'reg';      ;

eri1    pin                     ; "external error input 1
eri2    pin                     ; "external error input 2

checkb  node 114     istype 'reg';      "check first level

statb3  node 59      istype 'reg';      "cycle status first level bit 3
statb2  node 44      istype 'reg';      "cycle status first level bit 2
statb1  node 63      istype 'reg';      "cycle status first level bit 1
statb0  node 39      istype 'reg';      "cycle status first level bit 0

beb3    node 26      istype 'reg';      "byte enable first level da(31:24)
beb2    node 74      istype 'reg';      "byte enable first level da(23:16)
beb1    node 58      istype 'reg';      "byte enable first level da(15: 8)
beb0    node 45      istype 'reg';      "byte enable first level da( 7: 0)
```

```
pb3     node 9      istype 'reg';     "parity bit first level da(31:24)
pb2     node 71     istype 'reg';     "parity bit first level da(23:16)
pb1     node 64     istype 'reg';     "parity bit first level da(15: 8)
pb0     node 38     istype 'reg';     "parity bit first level da( 7: 0)


db31    node 25     istype 'reg';     "data bit first level
db30    node 57     istype 'reg';     "data bit first level
db29    node 46     istype 'reg';     "data bit first level
db28    node 28     istype 'reg';     "data bit first level
db27    node 10     istype 'reg';     "data bit first level
db26    node 65     istype 'reg';     "data bit first level
db25    node 37     istype 'reg';     "data bit first level
db24    node 56     istype 'reg';     "data bit first level

db23    node 47     istype 'reg';     "data bit first level
db22    node 66     istype 'reg';     "data bit first level
db21    node 86     istype 'reg';     "data bit first level
db20    node 55     istype 'reg';     "data bit first level
db19    node 92     istype 'reg';     "data bit first level
db18    node 85     istype 'reg';     "data bit first level
db17    node 80     istype 'reg';     "data bit first level
db16    node 79     istype 'reg';     "data bit first level

db15    node 98     istype 'reg';     "data bit first level
db14    node 84     istype 'reg';     "data bit first level
db13    node 90     istype 'reg';     "data bit first level
db12    node 91     istype 'reg';     "data bit first level
db11    node 104    istype 'reg';     "data bit first level
db10    node 89     istype 'reg';     "data bit first level
db9     node 78     istype 'reg';     "data bit first level
db8     node 116    istype 'reg';     "data bit first level

db7     node 83     istype 'reg';     "data bit first level
db6     node 97     istype 'reg';     "data bit first level
db5     node 77     istype 'reg';     "data bit first level
db4     node 88     istype 'reg';     "data bit first level
db3     node 82     istype 'reg';     "data bit first level
db2     node 73     istype 'reg';     "data bit first level
db1     node 72     istype 'reg';     "data bit first level
db0     node 69     istype 'reg';     "data bit first level

statc3  node 7      istype 'reg';     "cycle status second level bit 3
statc2  node 60     istype 'reg';     "cycle status second level bit 2
statc1  node 43     istype 'reg';     "cycle status second level bit 1
statc0  node 62     istype 'reg';     "cycle status second level bit 0

bec3    node 102    istype 'reg';     "byte enable second level da(31:24)
bec2    node 103    istype 'reg';     "byte enable second level da(23:16)
bec1    node 110    istype 'reg';     "byte enable second level da(15: 8)
bec0    node 109    istype 'reg';     "byte enable second level da( 7: 0)

dc11    node 111    istype 'reg';     "parity second level
dc10    node 105    istype 'reg';     "parity second level
dc9     node 99     istype 'reg';     "parity second level
```

```
dc8      node 95      istype 'reg';      "parity second level

dc7      node 113     istype 'reg';      "parity second level
dc6      node 108     istype 'reg';      "parity second level
dc5      node 101     istype 'reg';      "parity seceon level
dc4      node 94      istype 'reg';      "parity second level
dc3      node 112     istype 'reg';      "parity second level
dc2      node 106     istype 'reg';      "parity second level
dc1      node 100     istype 'reg';      "parity second level
dc0      node 93      istype 'reg';      "parity second level

err      node 70      istype 'reg, buffer';    "parity error;
errflag pin 14        istype 'reg_t, buffer'; "parity error; output pin
                                               "also output om da8

rx               node istype 'reg, buffer';    "used to clear error flag
ry               node istype 'reg, buffer';    "used to clear error flag

po0      node 96 istype 'reg, buffer'; "parity out da(2:0)
po1      node 87 istype 'reg, buffer'; "parity out da(5:3)
po2      node 75 istype 'reg, buffer'; "parity out da(7:6)

po3      node 81  istype 'reg, buffer'; "parity out da(10:8)
po4      node 107 istype 'reg, buffer'; "parity out da(13:11)
po5      node 115 istype 'reg, buffer'; "parity out da(15:14)


c, z, x = .C., .Z., .X.;

equations;

da15.c = clk;
da14.c = clk;
da13.c = clk;
da12.c = clk;
da11.c = clk;
da10.c = clk;
da9.c = clk;
da8.c = clk;

da7.c = clk;
da6.c = clk;
da5.c = clk;
da4.c = clk;
da3.c = clk;
da2.c = clk;
da1.c = clk;
da0.c = clk;

checkb.c = clk;

statb3.c = clk;
statb2.c = clk;
statb1.c = clk;
statb0.c = clk;
```

```
beb3.c = clk;
beb2.c = clk;
beb1.c = clk;
beb0.c = clk;

pb3.c = clk;
pb2.c = clk;
pb1.c = clk;
pb0.c = clk;

db31.c = clk;
db30.c = clk;
db29.c = clk;
db28.c = clk;
db27.c = clk;
db26.c = clk;
db25.c = clk;
db24.c = clk;

db23.c = clk;
db22.c = clk;
db21.c = clk;
db20.c = clk;
db19.c = clk;
db18.c = clk;
db17.c = clk;
db16.c = clk;

db15.c = clk;
db14.c = clk;
db13.c = clk;
db12.c = clk;
db11.c = clk;
db10.c = clk;
db9.c = clk;
db8.c = clk;

db7.c = clk;
db6.c = clk;
db5.c = clk;
db4.c = clk;
db3.c = clk;
db2.c = clk;
db1.c = clk;
db0.c = clk;

statc3.c = clk;
statc2.c = clk;
statc1.c = clk;
statc0.c = clk;
bec3.c = clk;
bec2.c = clk;
bec1.c = clk;
bec0.c = clk;

dcl1.c = clk;
```

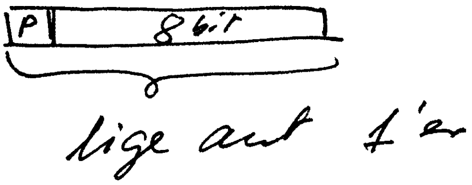
```
  dc10.c = clk;
  dc9.c = clk;
  dc8.c = clk;
  dc7.c = clk;
  dc6.c = clk;
  dc5.c = clk;
  dc4.c = clk;
  dc3.c = clk;
  dc2.c = clk;
  dc1.c = clk;
  dc0.c = clk;

  pchk.c = clk;

  err.c = clk;
  errflag.c = clk;

  rx.c = clk;
  ry.c = clk;

  da15.oe = read & cs;
  da14.oe = read & cs;
  da13.oe = read & cs;
  da12.oe = read & cs;
  da11.oe = read & cs;
  da10.oe = read & cs;
  da9.oe  = read & cs;
  da8.oe  = read & cs;

  da7.oe = read & cs;
  da6.oe = read & cs;
  da5.oe = read & cs;
  da4.oe = read & cs;
  da3.oe = read & cs;
  da2.oe = read & cs;
  da1.oe = read & cs;
  da0.oe = read & cs;

  rx.d   = read & cs;
  ry.d   = rx.q;

  "################################################################################
#
  "                                first level
  "################################################################################
#

  checkb :=(drdcen # drack # mast & !jmpc1 & jmpc0;). ~~cs.read~~   OK

  statb3 := stata3;
  statb2 := stata2;
  statb1 := stata1;
  statb0 := stata0;

  beb3 := bea3;
  beb2 := bea2;
  beb1 := bea1;
  beb0 := bea0;
  pb3 := pa3;
  pb2 := pa2;
  pb1 := pa1;
  pb0 := pa0;

  db31  := da31;
  db30  := da30;
  db29  := da29;
  db28  := da28;
  db27  := da27;
  db26  := da26;
  db25  := da25;
  db24  := da24;

  db23  := da23;
  db22  := da22;
  db21  := da21;
  db20  := da20;
  db19  := da19;
  db18  := da18;
  db17  := da17;
  db16  := da16;

  db15  := da15.pin;
  db14  := da14.pin;
  db13  := da13.pin;
  db12  := da12.pin;
  db11  := da11.pin;
  db10  := da10.pin;
  db9  :=   da9.pin;
  db8  :=   da8.pin;

  db7  :=   da7.pin;
  db6  :=   da6.pin;
  db5  :=   da5.pin;
  db4  :=   da4.pin;
  db3  :=   da3.pin;
  db2  :=   da2.pin;
  db1  :=   da1.pin;
  db0  :=   da0.pin;

  pchk := pchki; "parity error from ethernet controller

  "############################################################################
#
  "                            second level
  "############################################################################
#

  statc3 := statb3;
  statc2 := statb2;
  statc1 := statb1;
  statc0 := statb0;

  bec3 := beb3 & checkb;
  bec2 := beb2 & checkb;
  bec1 := beb1 & checkb;
```

```
bec0 := beb0 & checkb;

dc11:= pb3  $ db31 $ db30;
dc10:= db29 $ db28 $ db27;
dc9 := db26 $ db25 $ db24;

dc8 := pb2  $ db23 $ db22;
dc7 := db21 $ db20 $ db19;
dc6 := db18 $ db17 $ db16;

dc5 := pb1  $ db15 $ db14;
dc4 := db13 $ db12 $ db11;
dc3 := db10 $ db9  $ db8 ;

dc2 := pb0  $ db7  $ db6 ;
dc1 := db5  $ db4  $ db3 ;
dc0 := db2  $ db1  $ db0 ;

"##########################################################################
"                              third level
"##########################################################################

da3.d   = (dc11 $ dc10 $ dc9) & bec3 & !err.q & !errflag.q
        # da3.q & (err.q # errflag.q);
da2.d   = (dc8  $ dc7 $  dc6) & bec2 & !err.q & !errflag.q
        # da2.q & (err.q # errflag.q);
da1.d   = (dc5  $ dc4 $  dc3) & bec2 & !err.q & !errflag.q
        # da1.q & (err.q # errflag.q);
da0.d   = (dc2  $ dc1 $  dc0) & bec0 & !err.q & !errflag.q
        # da0.q & (err.q # errflag.q);

err.d   = (dc11 $ dc10 $ dc9) & bec3
        # (dc8  $ dc7  $ dc6) & bec2
        # (dc5  $ dc4  $ dc3) & bec1
        # (dc2  $ dc1  $ dc0) & bec0;

errflag.t =
        !errflag.q & (err.q # eri1 # eri2
       # pchk  # mast & jmpc1 & jmpc0)
       # errflag.q & !rx.q & ry.q; "clear condition

"errorflag is cleared after read.

da4.d   = err.q & !errflag.q # errflag.q;

da5.d   = statc3 & !err.q & !errflag.q # da5.q & (err.q # errflag.q);
da6.d   = statc2 & !err.q & !errflag.q # da6.q & (err.q # errflag.q);
da7.d   = statc1 & !err.q & !errflag.q # da7.q & (err.q # errflag.q);
da8.d   = statc0 & !err.q & !errflag.q # da8.q & (err.q # errflag.q);

da9.d   = pchk   & !err.q & !errflag.q # da5.q & (err.q # errflag.q);
da10.d  = eri1   & !err.q & !errflag.q # da6.q & (err.q # errflag.q);
da11.d  = eri2   & !err.q & !errflag.q # da7.q & (err.q # errflag.q);
da12.d  = jmpc0  & !err.q & !errflag.q # da8.q & (err.q # errflag.q);
da13.d  = jmpc1  & !err.q & !errflag.q # da8.q & (err.q # errflag.q);
da14.d  = mast   & !err.q & !errflag.q # da8.q & (err.q # errflag.q);
```

data bør holdes når cs.read !

```
"generation of parity out

po0.d    = da2.q $ da1.q  $ da0.q;
po1.d    = da5.q $ da4.q  $ da3.q;
po2.d    = da7.q $ da6.q;

pa0.d    = po2.q $  po1.q $ po0.q;

po3.d    = da10.q $ da9.q  $ da8.q;
po4.d    = da13.q $ da12.q  $ da11.q;
po5.d    = da15.q $ da14.q;

pa1.d    = po5.q $  po4.q $ po3.q;

end par1
```

```
-----------------------------------------------------

User pre-assignments:
    clk                     PIN 15
    read                    PIN 17
    cs                      PIN 54
    drack                   PIN 10
    drdcen                  PIN 9
    mast                    PIN 50
    jmpc1                   PIN 20
    jmpc0                   PIN 51
    stata3                  PIN 38
    stata2                  PIN 44
    stata1                  PIN 3
    stata0                  PIN 47
    bea3                    PIN 55
    bea2                    PIN 7
    bea1                    PIN 39
    bea0                    PIN 41
    pa3                     PIN 62
    pa2                     PIN 4
    da31                    PIN 56
    da30                    PIN 28
    da29                    PIN 40
    da28                    PIN 59
    da27                    PIN 66
    da26                    PIN 16
    da25                    PIN 46
    da24                    PIN 25
    da23                    PIN 37
    da22                    PIN 49
    da21                    PIN 57
    da20                    PIN 26
    da19                    PIN 43
    da18                    PIN 58
    da17                    PIN 65
    da16                    PIN 64
    pchki                   PIN 45
    eri1                    PIN 63
    eri2                    PIN 60
    pa1                     PIN 48
    pa0                     PIN 67
    da15                    PIN 24
    da14                    PIN 31
    da13                    PIN 30
    da12                    PIN 29
    da11                    PIN 13
    da10                    PIN 12
    da9                     PIN 11
    da8                     PIN 22
    da7                     PIN 23
    da6                     PIN 33
    da5                     PIN 32
    da4                     PIN 36
    da3                     PIN 21
    da2                     PIN 6
```



```
    da1                      PIN 5
    da0                      PIN 2
    errflag                  PIN 14
    checkb                   NODE 114
    statb3                   NODE 59
    statb2                   NODE 44
    statb1                   NODE 63
    statb0                   NODE 39
    beb3                     NODE 26
    beb2                     NODE 74
'beb2' has a direct connection from pin 'bea2'.
    beb1                     NODE 58
    beb0                     NODE 45
    pb3                      NODE 9
    pb2                      NODE 71
'pb2' has a direct connection from pin 'pa2'.
    pb1                      NODE 64
    pb0                      NODE 38
    db31                     NODE 25
    db30                     NODE 57
    db29                     NODE 46
    db28                     NODE 28
    db27                     NODE 10
    db26                     NODE 65
    db25                     NODE 37
    db24                     NODE 56
    db23                     NODE 47
    db22                     NODE 66
    db21                     NODE 86
    db20                     NODE 55
    db19                     NODE 92
    db18                     NODE 85
    db17                     NODE 80
    db16                     NODE 79
    db15                     NODE 98
    db14                     NODE 84
    db13                     NODE 90
'db13' has a direct connection from pin 'da13'.
    db12                     NODE 91
'db12' has a direct connection from pin 'da12'.
    db11                     NODE 104
    db10                     NODE 89
    db9                      NODE 78
'db9' has a direct connection from pin 'da9'.
    db8                      NODE 116
    db7                      NODE 83
'db7' has a direct connection from pin 'da7'.
    db6                      NODE 97
    db5                      NODE 77
    db4                      NODE 88
    db3                      NODE 82
    db2                      NODE 73
'db2' has a direct connection from pin 'da2'.
    db1                      NODE 72
'db1' has a direct connection from pin 'da1'.
    db0                      NODE 69
```

```
'db0' has a direct connection from pin 'da0'.
    statc3                      NODE 7
    statc2                      NODE 60
    statc1                      NODE 43
    statc0                      NODE 62
    bec3                        NODE 102
    bec2                        NODE 103
    bec1                        NODE 110
    bec0                        NODE 109
    dc11                        NODE 111
    dc10                        NODE 105
    dc9                         NODE 99
    dc8                         NODE 95
    dc7                         NODE 113
    dc6                         NODE 108
    dc5                         NODE 101
    dc4                         NODE 94
    dc3                         NODE 112
    dc2                         NODE 106
    dc1                         NODE 100
    dc0                         NODE 93
    err                         NODE 70
    po0                         NODE 96
    po1                         NODE 87
    po2                         NODE 75
    po3                         NODE 81
    po4                         NODE 107
    po5                         NODE 115
Warning 4435: Cannot fit signal po5
        H9 : Cannot route all inputs.
Warning 4442: -preassign TRY is on ... removing pre-assignment


---------------------------------------------------
Results of fitting par.tt2 into MACH220.
---------------------------------------------------

---------- Dedicated Inputs --------
PIN:17  --> read
PIN:20  --> jmpc1
PIN:51  --> jmpc0
PIN:54  --> cs

---------- Clock Inputs ------------
PIN:15  --> clk
PIN:16  --> da26
PIN:49  --> da22
PIN:50  --> mast


------------------------------------
       Resources of block A
------------------------------------
Inputs routed to block:  read cs statb3 bec3 bec2 bec1 bec0 dc11 dc10 dc9
dc8 dc7 dc6 dc5 dc4 dc3 dc2 dc1 dc0 rx.Q err.Q errflag.Q da2.Q da1.Q da0.Q
```

```
Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : read !cs
   OE2   : <none>

Resource allocation to macro cells:

  A0    (PIN :2   ) -->
         PIN INPUT          : da0.PIN
         PIN OUTPUT         : da0.D (pts=6)
         PRODUCT TERMS      : LOCAL
  A1    (NODE:69 ) -->
         INTERNAL FEEDBACK : db0.REG (direct connection from pin above)
         PRODUCT TERMS      : UP1
  A2    (PIN :3   ) -->
         PIN INPUT          : stata1
         PRODUCT TERMS      : DOWN
  A3    (NODE:70 ) -->
         INTERNAL FEEDBACK : err.D (pts=16)
         PRODUCT TERMS      : LOCAL
  A4    (PIN :4   ) -->
         PIN INPUT          : pa2
         INTERNAL FEEDBACK : ry.D (pts=1)
         PRODUCT TERMS      : UP1
  A5    (NODE:71 ) -->
         INTERNAL FEEDBACK : pb2.REG (direct connection from pin above)
         PRODUCT TERMS      : UP2
  A6    (PIN :5   ) -->
         PIN INPUT          : da1.PIN
         PIN OUTPUT         : da1.D (pts=6)
         PRODUCT TERMS      : UP2
  A7    (NODE:72 ) -->
         INTERNAL FEEDBACK : db1.REG (direct connection from pin above)
         PRODUCT TERMS      : UP1
  A8    (PIN :6   ) -->
         PIN INPUT          : da2.PIN
         PIN OUTPUT         : da2.D (pts=6)
         PRODUCT TERMS      : UP2
  A9    (NODE:73 ) -->
         INTERNAL FEEDBACK : db2.REG (direct connection from pin above)
         PRODUCT TERMS      : UP1
  A10   (PIN :7   ) -->
         PIN INPUT          : bea2
         INTERNAL FEEDBACK : statc3.REG (pts=1)
         PRODUCT TERMS      : UP2
  A11   (NODE:74 ) -->
         INTERNAL FEEDBACK : beb2.REG (direct connection from pin above)
         PRODUCT TERMS      : UP1

------------------------------------
       Block A resource summary:

                USED    FREE    UTILIZATION
  product terms  36      12      75.0%
  macro cells     12       0      100.0%
  pins             6       0      100.0%
------------------------------------



------------------------------------
       Resources of block B
------------------------------------
Inputs routed to block:  read cs mast jmpc1 jmpc0 pa3 da27 da17 da16 pchk
eri1 eri2 rx.Q da5.PIN err.Q errflag.Q ry.Q da5.Q da6.Q da7.Q da14.Q

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : read !cs
   OE2   : <none>

Resource allocation to macro cells:

  B0   (PIN :14 ) -->
        PIN OUTPUT           : errflag.T (pts=6)
        PRODUCT TERMS        : LOCAL
  B1   (NODE:75 ) -->
        INTERNAL FEEDBACK : po2.D (pts=2)
        PRODUCT TERMS        : LOCAL
  B2   (PIN :13 ) -->
        PIN INPUT            : da11.PIN
        PIN OUTPUT           : da11.D (pts=3)
        PRODUCT TERMS        : UP2
  B3   (NODE:76 ) -->
        INTERNAL FEEDBACK : po5.D (pts=1)
        PRODUCT TERMS        : LOCAL
  B4   (PIN :12 ) -->
        PIN INPUT            : da10.PIN
        PIN OUTPUT           : da10.D (pts=3)
        PRODUCT TERMS        : UP2
  B5   (NODE:77 ) -->
        INTERNAL FEEDBACK : db5.REG (pts=1)
        PRODUCT TERMS        : UP1
  B6   (PIN :11 ) -->
        PIN INPUT            : da9.PIN
        PIN OUTPUT           : da9.D (pts=3)
        PRODUCT TERMS        : UP1
  B7   (NODE:78 ) -->
        INTERNAL FEEDBACK : db9.REG (direct connection from pin above)
        PRODUCT TERMS        : UP1
  B8   (PIN :10 ) -->
        PIN INPUT            : drack
        INTERNAL FEEDBACK : db27.REG (pts=1)
        PRODUCT TERMS        : LOCAL
  B9   (NODE:79 ) -->
        INTERNAL FEEDBACK : db16.REG (pts=1)
        PRODUCT TERMS        : LOCAL
  B10  (PIN :9   ) -->
```

```
        PIN INPUT           : drdcen
        INTERNAL FEEDBACK : pb3.REG (pts=1)
        PRODUCT TERMS       : LOCAL
  B11   (NODE:80 ) -->
        INTERNAL FEEDBACK : db17.REG (pts=1)
        PRODUCT TERMS       : LOCAL

------------------------------------
       Block B resource summary:

                 USED     FREE     UTILIZATION
  product terms  23        25        47.9%
  macro cells    12         0       100.0%
  pins            6         0       100.0%
------------------------------------



------------------------------------
       Resources of block C
------------------------------------
Inputs routed to block:  read cs bea3 da31 da21 da18 statc1 statc0 bec3
dc11 dc10 dc9 da14.PIN da3.PIN err.Q errflag.Q da3.Q da7.Q da8.Q da10.Q
da9.Q

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : read !cs
   OE2   : <none>

Resource allocation to macro cells:

  C0    (PIN :21 ) -->
        PIN INPUT           : da3.PIN
        PIN OUTPUT          : da3.D (pts=6)
        PRODUCT TERMS       : LOCAL
  C1    (NODE:81 ) -->
        INTERNAL FEEDBACK : po3.D (pts=4)
        PRODUCT TERMS       : LOCAL
  C2    (PIN :22 ) -->
        PIN INPUT           : da8.PIN
        PIN OUTPUT          : da8.D (pts=3)
        PRODUCT TERMS       : UP2
  C3    (NODE:82 ) -->
        INTERNAL FEEDBACK : db3.REG (pts=1)
        PRODUCT TERMS       : UP1
  C4    (PIN :23 ) -->
        PIN INPUT           : da7.PIN
        PIN OUTPUT          : da7.D (pts=3)
        PRODUCT TERMS       : UP1
  C5    (NODE:83 ) -->
        INTERNAL FEEDBACK : db7.REG (direct connection from pin above)
        PRODUCT TERMS       : UP1
  C6    (PIN :24 ) -->
```

```
        PIN INPUT           : da15.PIN
        PIN OUTPUT          : da15.D (pts=1)
        PRODUCT TERMS       : LOCAL
  C7   (NODE:84 ) -->
        INTERNAL FEEDBACK : db14.REG (pts=1)
        PRODUCT TERMS       : LOCAL
  C8   (PIN :25 ) -->
        PIN INPUT           : da24
        INTERNAL FEEDBACK : db31.REG (pts=1)
        PRODUCT TERMS       : LOCAL
  C9   (NODE:85 ) -->
        INTERNAL FEEDBACK : db18.REG (pts=1)
        PRODUCT TERMS       : LOCAL
  C10  (PIN :26 ) -->
        PIN INPUT           : da20
        INTERNAL FEEDBACK : beb3.REG (pts=1)
        PRODUCT TERMS       : LOCAL
  C11  (NODE:86 ) -->
        INTERNAL FEEDBACK : db21.REG (pts=1)
        PRODUCT TERMS       : LOCAL

------------------------------------
       Block C resource summary:

                 USED     FREE    UTILIZATION
  product terms  23       25       47.9%
  macro cells    12        0       100.0%
  pins            6        0       100.0%
------------------------------------



------------------------------------
       Resources of block D
------------------------------------
Inputs routed to block:  read cs mast jmpc1 jmpc0 da28 da19 statc3 statc2
da10.PIN da4.PIN err.Q errflag.Q da3.Q da5.Q da6.Q da8.Q da4.Q

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : read !cs
   OE2   : <none>

Resource allocation to macro cells:

  D0   (PIN :33 ) -->
        PIN INPUT           : da6.PIN
        PIN OUTPUT          : da6.D (pts=3)
        PRODUCT TERMS       : LOCAL
  D1   (NODE:87 ) -->
        INTERNAL FEEDBACK : po1.D (pts=4)
        PRODUCT TERMS       : LOCAL
  D2   (PIN :32 ) -->
        PIN INPUT           : da5.PIN
       PIN OUTPUT          : da5.D (pts=3)
       PRODUCT TERMS       : LOCAL
 D3   (NODE:88 ) -->
       INTERNAL FEEDBACK : db4.REG (pts=1)
       PRODUCT TERMS       : LOCAL
 D4   (PIN :31 ) -->
       PIN INPUT           : da14.PIN
       PIN OUTPUT          : da14.D (pts=3)
       PRODUCT TERMS       : LOCAL
 D5   (NODE:89 ) -->
       INTERNAL FEEDBACK : db10.REG (pts=1)
       PRODUCT TERMS       : LOCAL
 D6   (PIN :30 ) -->
       PIN INPUT           : da13.PIN
       PIN OUTPUT          : da13.D (pts=3)
       PRODUCT TERMS       : LOCAL
 D7   (NODE:90 ) -->
       INTERNAL FEEDBACK : db13.REG (direct connection from pin above)
       PRODUCT TERMS       : NOT USED
 D8   (PIN :29 ) -->
       PIN INPUT           : da12.PIN
       PIN OUTPUT          : da12.D (pts=3)
       PRODUCT TERMS       : LOCAL
 D9   (NODE:91 ) -->
       INTERNAL FEEDBACK : db12.REG (direct connection from pin above)
       PRODUCT TERMS       : NOT USED
 D10  (PIN :28 ) -->
       PIN INPUT           : da30
       INTERNAL FEEDBACK : db28.REG (pts=1)
       PRODUCT TERMS       : LOCAL
 D11  (NODE:92 ) -->
       INTERNAL FEEDBACK : db19.REG (pts=1)
       PRODUCT TERMS       : LOCAL

------------------------------------
      Block D resource summary:

                USED    FREE    UTILIZATION
 product terms  23       25       47.9%
 macro cells     12        0      100.0%
 pins             6        0      100.0%
------------------------------------



------------------------------------
      Resources of block E
------------------------------------
Inputs routed to block:  read cs stata0 pa0 da25 pchki pb2 db23 db22 db13
db12 db11 db2 db1 db0 da15.PIN da6.PIN err.Q errflag.Q da2.Q da1.Q da0.Q

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : read !cs
   OE2   : <none>

Resource allocation to macro cells:

  E0    (PIN :36 ) -->
         PIN INPUT          : da4.PIN
         PIN OUTPUT         : da4.D (pts=2)
         PRODUCT TERMS      : LOCAL
  E1    (NODE:93 ) -->
         INTERNAL FEEDBACK : dc0.REG (pts=4)
         PRODUCT TERMS      : LOCAL
  E2    (PIN :37 ) -->
         PIN INPUT          : da23
         INTERNAL FEEDBACK : db25.REG (pts=1)
         PRODUCT TERMS      : LOCAL
  E3    (NODE:94 ) -->
         INTERNAL FEEDBACK : dc4.REG (pts=4)
         PRODUCT TERMS      : LOCAL
  E4    (PIN :38 ) -->
         PIN INPUT          : stata3
         INTERNAL FEEDBACK : pb0.REG (pts=1)
         PRODUCT TERMS      : LOCAL
  E5    (NODE:95 ) -->
         INTERNAL FEEDBACK : dc8.REG (pts=4)
         PRODUCT TERMS      : LOCAL
  E6    (PIN :39 ) -->
         PIN INPUT          : bea1
         INTERNAL FEEDBACK : statb0.REG (pts=1)
         PRODUCT TERMS      : LOCAL
  E7    (NODE:96 ) -->
         INTERNAL FEEDBACK : po0.D (pts=4)
         PRODUCT TERMS      : LOCAL
  E8    (PIN :40 ) -->
         PIN INPUT          : da29
         INTERNAL FEEDBACK : pchk.REG (pts=1)
         PRODUCT TERMS      : LOCAL
  E9    (NODE:97 ) -->
         INTERNAL FEEDBACK : db6.REG (pts=1)
         PRODUCT TERMS      : LOCAL
  E10   (PIN :41 ) -->
         PIN INPUT          : bea0
         INTERNAL FEEDBACK : rx.D (pts=1)
         PRODUCT TERMS      : LOCAL
  E11   (NODE:98 ) -->
         INTERNAL FEEDBACK : db15.REG (pts=1)
         PRODUCT TERMS      : LOCAL

------------------------------------
       Block E resource summary:

                 USED     FREE    UTILIZATION
  product terms  25       23       52.1%
  macro cells    12        0       100.0%
  pins            6        0       100.0%
------------------------------------
```

```
-----------------------------------
      Resources of block F
-----------------------------------
Inputs routed to block:  read cs stata2 bea0 da29 da23 checkb statb1 beb3
beb2 pb1 db26 db25 db24 db15 db14 db5 db4 db3 da11.PIN po5.Q po4.Q po3.Q

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : read !cs
   OE2   : <none>

Resource allocation to macro cells:

  F0   (PIN :48 ) -->
        PIN INPUT          : pa1
        PIN OUTPUT         : pa1.D (pts=4)
        PRODUCT TERMS      : LOCAL
  F1   (NODE:99 ) -->
        INTERNAL FEEDBACK : dc9.REG (pts=4)
        PRODUCT TERMS      : LOCAL
  F2   (PIN :47 ) -->
        PIN INPUT          : stata0
        INTERNAL FEEDBACK : db23.REG (pts=1)
        PRODUCT TERMS      : LOCAL
  F3   (NODE:100) -->
        INTERNAL FEEDBACK : dc1.REG (pts=4)
        PRODUCT TERMS      : LOCAL
  F4   (PIN :46 ) -->
        PIN INPUT          : da25
        INTERNAL FEEDBACK : db29.REG (pts=1)
        PRODUCT TERMS      : LOCAL
  F5   (NODE:101) -->
        INTERNAL FEEDBACK : dc5.REG (pts=4)
        PRODUCT TERMS      : LOCAL
  F6   (PIN :45 ) -->
        PIN INPUT          : pchki
        INTERNAL FEEDBACK : beb0.REG (pts=1)
        PRODUCT TERMS      : LOCAL
  F7   (NODE:102) -->
        INTERNAL FEEDBACK : bec3.REG (pts=1)
        PRODUCT TERMS      : LOCAL
  F8   (PIN :44 ) -->
        PIN INPUT          : stata2
        INTERNAL FEEDBACK : statb2.REG (pts=1)
        PRODUCT TERMS      : LOCAL
  F9   (NODE:103) -->
        INTERNAL FEEDBACK : bec2.REG (pts=1)
        PRODUCT TERMS      : LOCAL
  F10  (PIN :43 ) -->
        PIN INPUT          : da19
        INTERNAL FEEDBACK : statc1.REG (pts=1)
        PRODUCT TERMS      : LOCAL
  F11   (NODE:104) -->
         INTERNAL FEEDBACK : db11.REG (pts=1)
         PRODUCT TERMS     : LOCAL

------------------------------------
       Block F resource summary:

                 USED     FREE    UTILIZATION
  product terms  24       24       50.0%
  macro cells    12        0       100.0%
  pins            6        0       100.0%
------------------------------------



------------------------------------
       Resources of block G
------------------------------------
Inputs routed to block:  stata3 bea1 da30 da24 da20 checkb statb2 beb1 beb0
pb0 db29 db28 db27 db18 db17 db16 db7 db6 da13.Q da12.Q da11.Q

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : <none>
   OE2   : <none>

Resource allocation to macro cells:

  G0    (PIN :55 ) -->
         PIN INPUT         : bea3
         INTERNAL FEEDBACK : db20.REG (pts=1)
         PRODUCT TERMS     : LOCAL
  G1    (NODE:105) -->
         INTERNAL FEEDBACK : dc10.REG (pts=4)
         PRODUCT TERMS     : LOCAL
  G2    (PIN :56 ) -->
         PIN INPUT         : da31
         INTERNAL FEEDBACK : db24.REG (pts=1)
         PRODUCT TERMS     : LOCAL
  G3    (NODE:106) -->
         INTERNAL FEEDBACK : dc2.REG (pts=4)
         PRODUCT TERMS     : LOCAL
  G4    (PIN :57 ) -->
         PIN INPUT         : da21
         INTERNAL FEEDBACK : db30.REG (pts=1)
         PRODUCT TERMS     : LOCAL
  G5    (NODE:107) -->
         INTERNAL FEEDBACK : po4.D (pts=4)
         PRODUCT TERMS     : LOCAL
  G6    (PIN :58 ) -->
         PIN INPUT         : da18
         INTERNAL FEEDBACK : beb1.REG (pts=1)
         PRODUCT TERMS     : LOCAL
  G7    (NODE:108) -->
```

```
        INTERNAL FEEDBACK : dc6.REG (pts=4)
        PRODUCT TERMS     : LOCAL
  G8   (PIN :59 ) -->
        PIN INPUT         : da28
        INTERNAL FEEDBACK : statb3.REG (pts=1)
        PRODUCT TERMS     : LOCAL
  G9   (NODE:109) -->
        INTERNAL FEEDBACK : bec0.REG (pts=1)
        PRODUCT TERMS     : LOCAL
  G10  (PIN :60 ) -->
        PIN INPUT         : eri2
        INTERNAL FEEDBACK : statc2.REG (pts=1)
        PRODUCT TERMS     : LOCAL
  G11  (NODE:110) -->
        INTERNAL FEEDBACK : bec1.REG (pts=1)
        PRODUCT TERMS     : LOCAL

------------------------------------
       Block G resource summary:

                  USED     FREE     UTILIZATION
  product terms   24        24        50.0%
  macro cells     12         0        100.0%
  pins             6         0        100.0%
------------------------------------



------------------------------------
       Resources of block H
------------------------------------
Inputs routed to block:  read cs drack drdcen mast jmpc1 jmpc0 stata1 pa1
da26 da22 statb0 pb3 db31 db30 db21 db20 db19 db10 db9 db8 da8.PIN po2.Q
po1.Q po0.Q

Controls on block:

   RESET : <none>
   PRESET: <none>
   OE1   : read !cs
   OE2   : <none>

Resource allocation to macro cells:

  H0   (PIN :67 ) -->
        PIN INPUT         : pa0
        PIN OUTPUT        : pa0.D (pts=4)
        PRODUCT TERMS     : LOCAL
  H1   (NODE:111) -->
        INTERNAL FEEDBACK : dc11.REG (pts=4)
        PRODUCT TERMS     : LOCAL
  H2   (PIN :66 ) -->
        PIN INPUT         : da27
        INTERNAL FEEDBACK : db22.REG (pts=1)
        PRODUCT TERMS     : LOCAL
  H3   (NODE:112) -->
```

```
        INTERNAL FEEDBACK : dc3.REG (pts=4)
        PRODUCT TERMS     : LOCAL
  H4   (PIN :65 ) -->
        PIN INPUT         : da17
        INTERNAL FEEDBACK : db26.REG (pts=1)
        PRODUCT TERMS     : LOCAL
  H5   (NODE:113) -->
        INTERNAL FEEDBACK : dc7.REG (pts=4)
        PRODUCT TERMS     : LOCAL
  H6   (PIN :64 ) -->
        PIN INPUT         : da16
        INTERNAL FEEDBACK : pb1.REG (pts=1)
        PRODUCT TERMS     : LOCAL
  H7   (NODE:114) -->
        INTERNAL FEEDBACK : checkb.REG (pts=3)
        PRODUCT TERMS     : LOCAL
  H8   (PIN :63 ) -->
        PIN INPUT         : eri1
        INTERNAL FEEDBACK : statb1.REG (pts=1)
        PRODUCT TERMS     : LOCAL
  H9   (NODE:115) -->
        PRODUCT TERMS     : NOT USED
  H10  (PIN :62 ) -->
        PIN INPUT         : pa3
        INTERNAL FEEDBACK : statc0.REG (pts=1)
        PRODUCT TERMS     : LOCAL
  H11  (NODE:116) -->
        INTERNAL FEEDBACK : db8.REG (pts=1)
        PRODUCT TERMS     : LOCAL

------------------------------------
       Block H resource summary:

                USED    FREE    UTILIZATION
  product terms  25      23       52.1%
  macro cells    11       1       91.7%
  pins            6       0      100.0%
------------------------------------



------------------------------------
       CHIP resource summary:

                USED    FREE    UTILIZATION
  product terms 203     181       52.9%
  macro cells    95       1       99.0%
  pins           56       0      100.0%

------------------------------------

---------------------------------------------------
Routing results.
---------------------------------------------------

---------------------
```

```
 Routes for block A
---------------------
+--------------------------------------------+
|input -->           path            : signal |
+--------------------------------------------+
|  0  | --> internal route of G1      : dc10
|  1  | --> internal route of E1      : dc0
|  2  | --> internal route of E3      : dc4
|  3  | --> pin route of INPUT2       : read
|  4  | --> internal route of F5      : dc5
|  5  | --> internal route of F3      : dc1
|  6  | --> internal route of F1      : dc9
|  7  | --> pin route of INPUT7       : cs
|  8  | --> internal route of H3      : dc3
|  9  | --> internal route of A3      : err.Q
| 10  | --> internal route of E5      : dc8
| 11  | --> internal route of G3      : dc2
| 12  | --> internal route of G7      : dc6
| 13  | --> internal route of F7      : bec3
| 14  | --> internal route of G11     : bec1
| 15  | --> internal route of E10     : rx.Q
| 16  | --> internal route of H5      : dc7
| 17  | --> internal route of G8      : statb3
| 18  | --> internal route of A8      : da2.Q
| 19  | --> internal route of H1      : dc11
| 20  | --> internal route of B0      : errflag.Q
| 21  | --> internal route of F9      : bec2
| 22  | --> internal route of A0      : da0.Q
| 23  | -->
| 24  | --> internal route of G9      : bec0
| 25  | --> internal route of A6      : da1.Q

25 of 26 routes used (utilization = 96.2%)

---------------------
 Routes for block B
---------------------
+--------------------------------------------+
|input -->           path            : signal |
+--------------------------------------------+
|  0  | --> internal route of E8      : pchk
|  1  | --> internal route of E10     : rx.Q
|  2  | -->
|  3  | -->
|  4  | --> internal route of A3      : err.Q
|  5  | --> pin route of D2           : da5.PIN
|  6  | -->
|  7  | -->
|  8  | --> pin route of H2           : da27
|  9  | --> pin route of INPUT7       : cs
| 10  | --> internal route of D4      : da14.Q
| 11  | --> internal route of A4      : ry.Q
| 12  | --> pin route of H6           : da16
| 13  | --> pin route of H4           : da17
| 14  | --> internal route of B0      : errflag.Q
| 15  | --> pin route of INPUT3       : jmpc1
| 16 | --> pin route of INPUT6       : jmpc0
| 17 | -->
| 18 | --> internal route of C4      : da7.Q
| 19 | --> pin route of H10          : pa3
| 20 | --> pin route of H8           : eri1
| 21 | --> pin route of G10          : eri2
| 22 | --> internal route of D0      : da6.Q
| 23 | --> internal route of D2      : da5.Q
| 24 | --> pin route of INPUT2       : read
| 25 | --> pin route of INPUT5       : mast

21 of 26 routes used (utilization = 80.8%)

---------------------
 Routes for block C
---------------------
+-----------------------------------------------+
|input -->            path             : signal |
+-----------------------------------------------+
|  0 | --> internal route of G1      : dc10
|  1 | --> internal route of H10     : statc0
|  2 | --> internal route of B6      : da9.Q
|  3 | -->
|  4 | --> internal route of A3      : err.Q
|  5 | --> pin route of G2           : da31
|  6 | --> pin route of G4           : da21
|  7 | --> internal route of F7      : bec3
|  8 | -->
|  9 | --> pin route of INPUT7       : cs
| 10 | --> pin route of D4           : da14.PIN
| 11 | --> pin route of C0           : da3.PIN
| 12 | --> internal route of C0      : da3.Q
| 13 | --> internal route of C2      : da8.Q
| 14 | -->
| 15 | --> internal route of C4      : da7.Q
| 16 | -->
| 17 | --> pin route of INPUT2       : read
| 18 | --> pin route of G0           : bea3
| 19 | --> internal route of H1      : dc11
| 20 | --> internal route of B0      : errflag.Q
| 21 | -->
| 22 | --> internal route of B4      : da10.Q
| 23 | --> pin route of G6           : da18
| 24 | --> internal route of F10     : statc1
| 25 | --> internal route of F1      : dc9

21 of 26 routes used (utilization = 80.8%)

---------------------
 Routes for block D
---------------------
+-----------------------------------------------+
|input -->            path             : signal |
+-----------------------------------------------+
|  0 | --> internal route of C0      : da3.Q
|  1 | -->
|   2 | --> pin route of F10          : da19
|   3 | -->
|   4 | --> internal route of A3      : err.Q
|   5 | -->
|   6 | -->
|   7 | -->
|   8 | --> pin route of G8           : da28
|   9 | --> pin route of INPUT7       : cs
|  10 | -->
|  11 | --> internal route of E0      : da4.Q
|  12 | --> pin route of E0           : da4.PIN
|  13 | --> internal route of C2      : da8.Q
|  14 | --> internal route of B0      : errflag.Q
|  15 | --> pin route of INPUT3       : jmpc1
|  16 | --> internal route of G10     : statc2
|  17 | -->
|  18 | --> pin route of B4           : da10.PIN
|  19 | --> internal route of A10     : statc3
|  20 | --> pin route of INPUT6       : jmpc0
|  21 | -->
|  22 | --> internal route of D0      : da6.Q
|  23 | --> internal route of D2      : da5.Q
|  24 | --> pin route of INPUT2       : read
|  25 | --> pin route of INPUT5       : mast

18 of 26 routes used (utilization = 69.2%)

---------------------
 Routes for block E
---------------------
+----------------------------------------------+
|input -->            path            : signal |
+----------------------------------------------+
|   0 | --> internal route of F2      : db23
|   1 | -->
|   2 | --> internal route of F11     : db11
|   3 | -->
|   4 | --> internal route of A3      : err.Q
|   5 | --> internal route of A6      : da1.Q
|   6 | -->
|   7 | --> internal route of A8      : da2.Q
|   8 | --> internal route of A9      : db2
|   9 | --> pin route of INPUT7       : cs
|  10 | --> internal route of A7      : db1
|  11 | --> internal route of D9      : db12
|  12 | --> pin route of F4           : da25
|  13 | --> internal route of D7      : db13
|  14 | --> pin route of C6           : da15.PIN
|  15 | --> pin route of H0           : pa0
|  16 | -->
|  17 | --> pin route of D0           : da6.PIN
|  18 | --> internal route of A1      : db0
|  19 | --> internal route of H2      : db22
|  20 | --> internal route of B0      : errflag.Q
|  21 | --> pin route of F6           : pchki
|  22 | --> internal route of A0      : da0.Q
| 23 | --> pin route of F2          : stata0
| 24 | --> pin route of INPUT2      : read
| 25 | --> internal route of A5     : pb2

22 of 26 routes used (utilization = 84.6%)

---------------------
 Routes for block F
---------------------
+----------------------------------------------+
|input -->           path            : signal |
+----------------------------------------------+
|  0 | --> pin route of INPUT7      : cs
|  1 | --> internal route of C7     : db14
|  2 | --> internal route of A11    : beb2
|  3 | --> internal route of H7     : checkb
|  4 | -->
|  5 | -->
|  6 | --> pin route of B2          : da11.PIN
|  7 | --> internal route of C1     : po3.Q
|  8 | --> internal route of C3     : db3
|  9 | --> internal route of G5     : po4.Q
| 10 | --> internal route of E2     : db25
| 11 | --> internal route of H8     : statb1
| 12 | --> internal route of H6     : pb1
| 13 | --> pin route of E2          : da23
| 14 | --> internal route of E11    : db15
| 15 | --> internal route of D3     : db4
| 16 | --> internal route of B5     : db5
| 17 | --> pin route of INPUT2      : read
| 18 | --> internal route of H4     : db26
| 19 | -->
| 20 | --> pin route of F8          : stata2
| 21 | --> internal route of C10    : beb3
| 22 | --> internal route of G2     : db24
| 23 | --> pin route of E8          : da29
| 24 | --> pin route of E10         : bea0
| 25 | --> internal route of B3     : po5.Q

23 of 26 routes used (utilization = 88.5%)

---------------------
 Routes for block G
---------------------
+----------------------------------------------+
|input -->           path            : signal |
+----------------------------------------------+
|  0 | --> internal route of B2     : da11.Q
|  1 | --> internal route of F4     : db29
|  2 | --> internal route of D8     : da12.Q
|  3 | --> internal route of H7     : checkb
|  4 | -->
|  5 | -->
|  6 | --> internal route of E9     : db6
|  7 | --> internal route of E4     : pb0
|  8 | -->
|  9  | --> internal route of B8      : db27
| 10  | -->
| 11  | --> internal route of D6      : da13.Q
| 12  | --> internal route of B9      : db16
| 13  | --> internal route of C9      : db18
| 14  | --> pin route of E4           : stata3
| 15  | --> pin route of C8           : da24
| 16  | --> pin route of C10          : da20
| 17  | -->
| 18  | --> internal route of F6      : beb0
| 19  | --> internal route of D10     : db28
| 20  | --> internal route of G6      : beb1
| 21  | --> internal route of B11     : db17
| 22  | --> pin route of E6           : bea1
| 23  | --> pin route of D10          : da30
| 24  | --> internal route of C5      : db7
| 25  | --> internal route of F8      : statb2

21 of 26 routes used (utilization = 80.8%)

---------------------
 Routes for block H
---------------------
+-----------------------------------------------+
|input -->            path            : signal |
+-----------------------------------------------+
|  0  | --> pin route of INPUT4       : da22
|  1  | --> pin route of A2           : stata1
|  2  | --> internal route of G4      : db30
|  3  | --> pin route of INPUT5       : mast
|  4  | --> internal route of D1      : po1.Q
|  5  | --> internal route of B7      : db9
|  6  | --> internal route of H11     : db8
|  7  | --> pin route of INPUT7       : cs
|  8  | --> internal route of E6      : statb0
|  9  | --> pin route of B8           : drack
| 10  | --> pin route of B10          : drdcen
| 11  | -->
| 12  | --> pin route of C2           : da8.PIN
| 13  | --> internal route of G0      : db20
| 14  | --> internal route of B1      : po2.Q
| 15  | --> pin route of INPUT3       : jmpc1
| 16  | --> internal route of C11     : db21
| 17  | --> pin route of INPUT2       : read
| 18  | --> pin route of INPUT6       : jmpc0
| 19  | --> internal route of D11     : db19
| 20  | --> internal route of C8      : db31
| 21  | --> pin route of INPUT1       : da26
| 22  | --> internal route of D5      : db10
| 23  | --> internal route of E7      : po0.Q
| 24  | --> pin route of F0           : pa1
| 25  | --> internal route of B10     : pb3

25 of 26 routes used (utilization = 96.2%)

CHIP route summary: 176 of 208 routes used (utilization = 84.6%)
```

```
---------------------------------------------------


Routing Table:

BLOCK        SIGNAL
........        clk
ABCDEF.H        read
ABCDEF.H        cs
.......H        drack
.......H        drdcen
.B.D...H        mast
.B.D...H        jmpc1
.B.D...H        jmpc0
......G.        stata3
.....F..        stata2
.......H        stata1
....E...        stata0
..C.....        bea3
........        bea2
......G.        bea1
.....F..        bea0
.B......        pa3
........        pa2
.......H        pa1
....E...        pa0
..C.....        da31
......G.        da30
.....F..        da29
...D....        da28
.B......        da27
.......H        da26
....E...        da25
......G.        da24
.....F..        da23
.......H        da22
..C.....        da21
......G.        da20
...D....        da19
..C.....        da18
.B......        da17
.B......        da16
....E...        pchki
.B......        pchk
.B......        eri1
.B......        eri2
.....FG.        checkb
A.......        statb3
......G.        statb2
.....F..        statb1
.......H        statb0
.....F..        beb3
.....F..        beb2
......G.        beb1
......G.        beb0
.......H        pb3
```

```
....E...       pb2
.....F..       pb1
......G.       pb0
.......H       db31
.......H       db30
......G.       db29
......G.       db28
......G.       db27
.....F..       db26
.....F..       db25
.....F..       db24
....E...       db23
....E...       db22
.......H       db21
.......H       db20
.......H       db19
......G.       db18
......G.       db17
......G.       db16
.....F..       db15
.....F..       db14
....E...       db13
....E...       db12
....E...       db11
.......H       db10
.......H       db9
.......H       db8
......G.       db7
......G.       db6
.....F..       db5
.....F..       db4
.....F..       db3
....E...       db2
....E...       db1
....E...       db0
...D....       statc3
...D....       statc2
..C.....       statc1
..C.....       statc0
A.C.....       bec3
A.......       bec2
A.......       bec1
A.......       bec0
A.C.....       dc11
A.C.....       dc10
A.C.....       dc9
A.......       dc8
A.......       dc7
A.......       dc6
A.......       dc5
A.......       dc4
A.......       dc3
A.......       dc2
A.......       dc1
A.......       dc0
AB......       rx.Q
```

```
....E...       da15.PIN
..C.....       da14.PIN
........       da13.PIN
........       da12.PIN
.....F..       da11.PIN
...D....       da10.PIN
........       da9.PIN
.......H       da8.PIN
........       da7.PIN
....E...       da6.PIN
.B......       da5.PIN
...D....       da4.PIN
..C.....       da3.PIN
........       da2.PIN
........       da1.PIN
........       da0.PIN
ABCDE...       err.Q
ABCDE...       errflag.Q
..CD....       da3.Q
A...E...       da2.Q
A...E...       da1.Q
A...E...       da0.Q
.B......       ry.Q
.B.D....       da5.Q
.B.D....       da6.Q
.BC.....       da7.Q
..CD....       da8.Q
...D....       da4.Q
.......H       po2.Q
.......H       po1.Q
.......H       po0.Q
..C.....       da10.Q
..C.....       da9.Q
......G.       da13.Q
......G.       da12.Q
......G.       da11.Q
.B......       da14.Q
.....F..       po5.Q
.....F..       po4.Q
.....F..       po3.Q

Current partitioning requires 176 routes.
----------------------------------------------------
$DEVICE MACH220A fit par.tt3
$PINS 56 clk:15 read:17 cs:54 drack:10 drdcen:9 mast:50 jmpc1:20 jmpc0:51
stata3:38 stata2:44 stata1:3 stata0:47 bea3:55 bea2:7 bea1:39 bea0:41
pa3:62 pa2:4 da31:56 da30:28 da29:40 da28:59 da27:66 da26:16 da25:46 da24:25
da23:37 da22:49 da21:57 da20:26 da19:43 da18:58 da17:65 da16:64 pchki:45
eri1:63 eri2:60 errflag+:14 da0+:2 da1+:5 da2+:6 da3+:21 da8+:22 da7+:23
da6+:33 da5+:32 da12+:29 da13+:30 da14+:31 da11+:13 da10+:12 da9+:11 pa1+:48
pa0+:67 da4+:36 da15+:24
$NODES 75 pchk:40 db0:69 db1:72 db2:73 db3:82 db4:88 db5:77 db6:97
db7:83 db8:116 db9:78 db10:89 db11:104 db12:91 db13:90 db14:84 db15:98
db16:79 db17:80 db18:85 db19:92 db20:55 db21:86 db22:66 db23:47
db24:56 db25:37 db26:65 db27:10 db28:28 db29:46 db30:57 db31:25
pb0:38 pb1:64 pb2:71 pb3:9 beb0:45 beb1:58 beb2:74 beb3:26 statb0:39
```

```
statb1:63 statb2:44 statb3:59 err:70 checkb:114 dc0:93 dc1:100 dc2:106
dc3:112 dc4:94 dc5:101 dc6:108 dc7:113 dc8:95 dc9:99 dc10:105 dc11:111
po4:107 po3:81 po1:87 po0:96 bec2:103 bec0:109 bec3:102 rx:41 po2:75
bec1:110 po5:76 ry:4 statc0:62 statc1:43 statc2:60 statc3:7
----------------------------------------------------
```