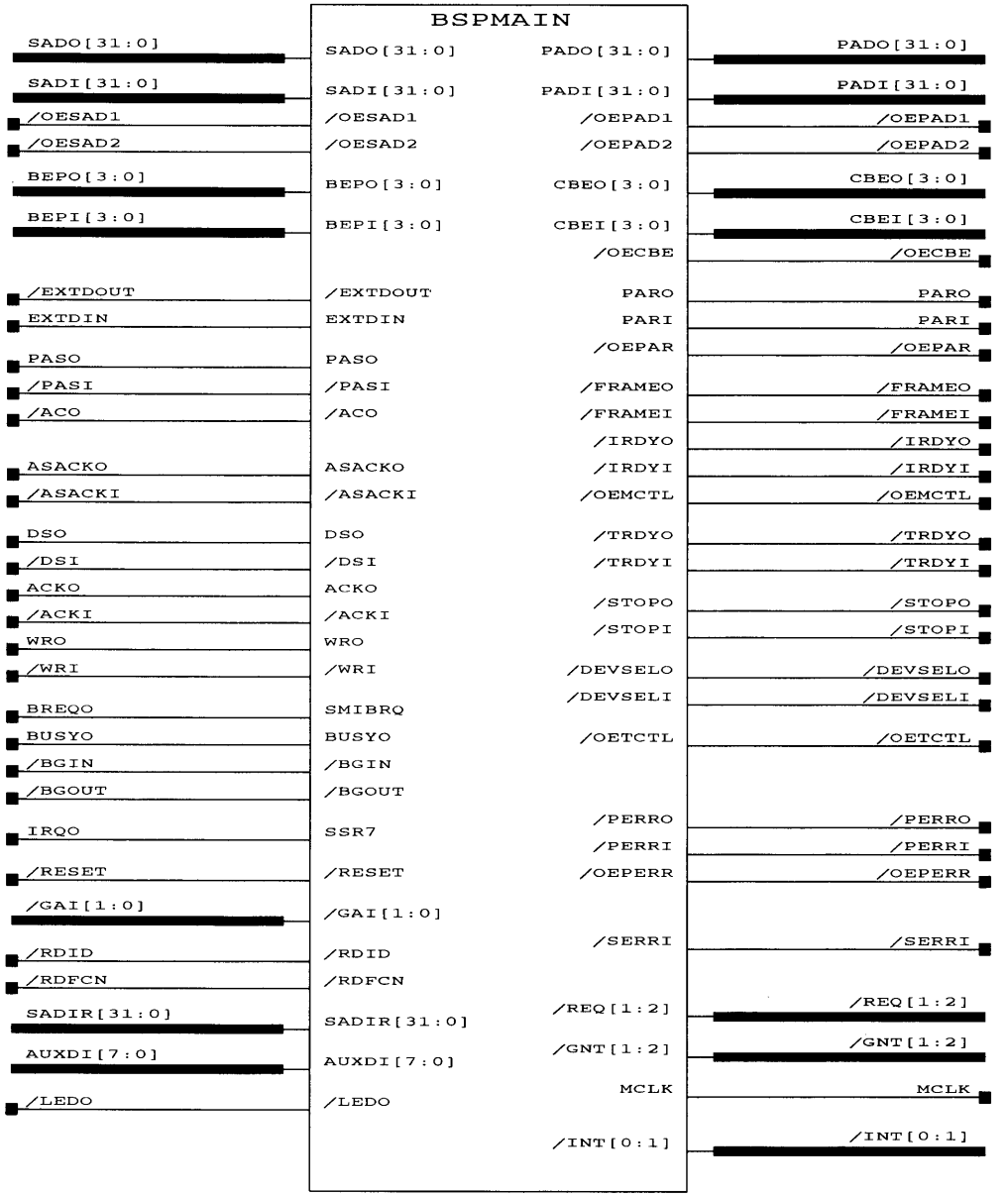
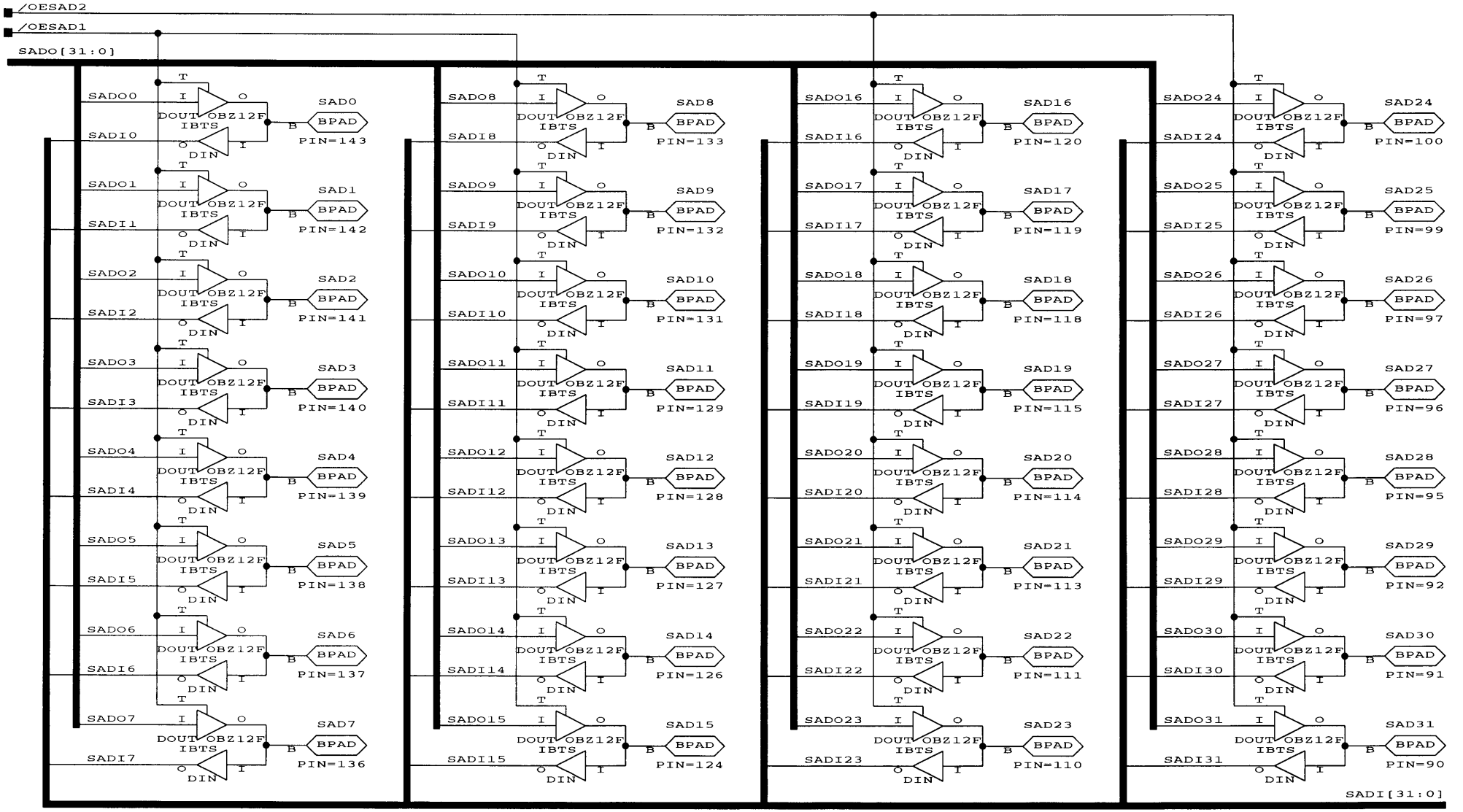
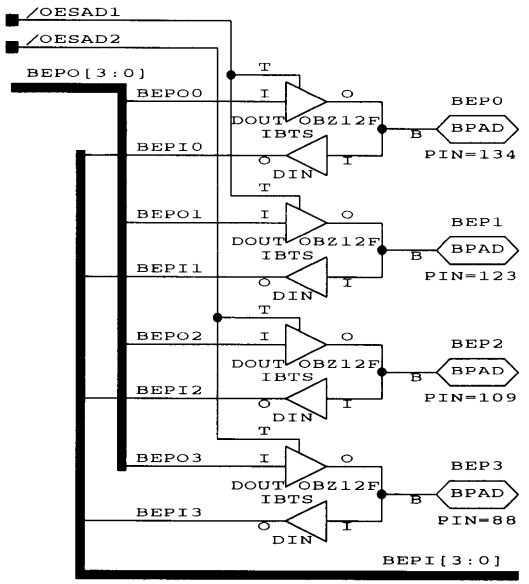
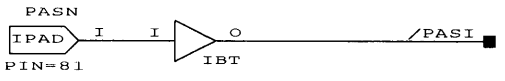


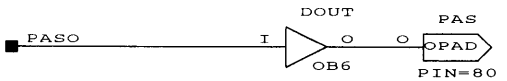

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970917 | BSP301-1 | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: bsp301 | Page: 1 of 6 |

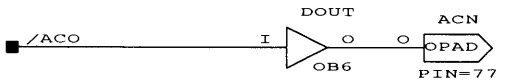


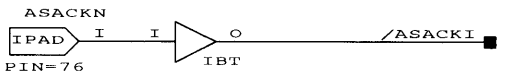

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI ADDR/DATA TRANSCEIVERS |
| Issue 2 | | |
| Issue 3 | | File: bsp301 Page: 2 of 6 |

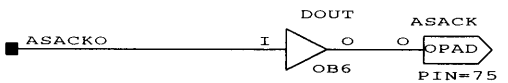


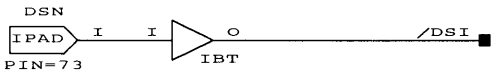



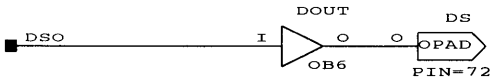



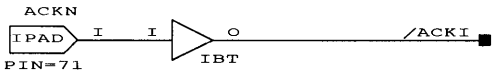



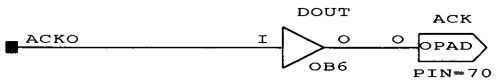



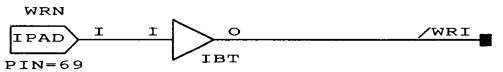



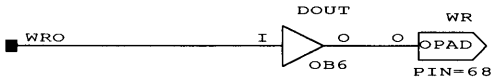



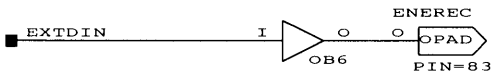



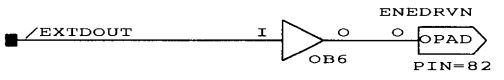



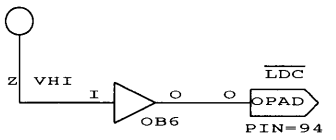



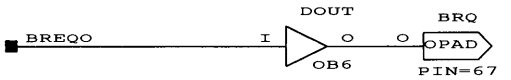



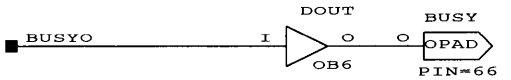



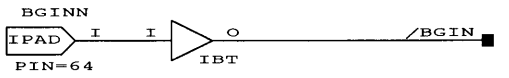



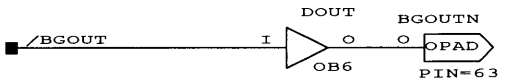



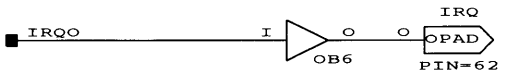



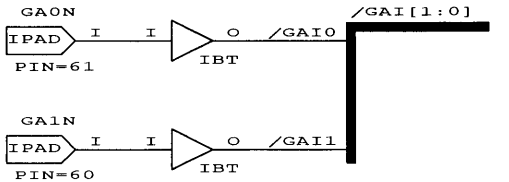



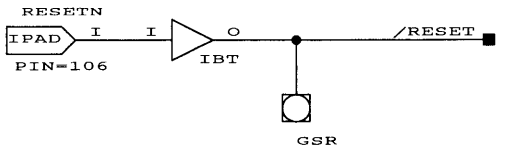



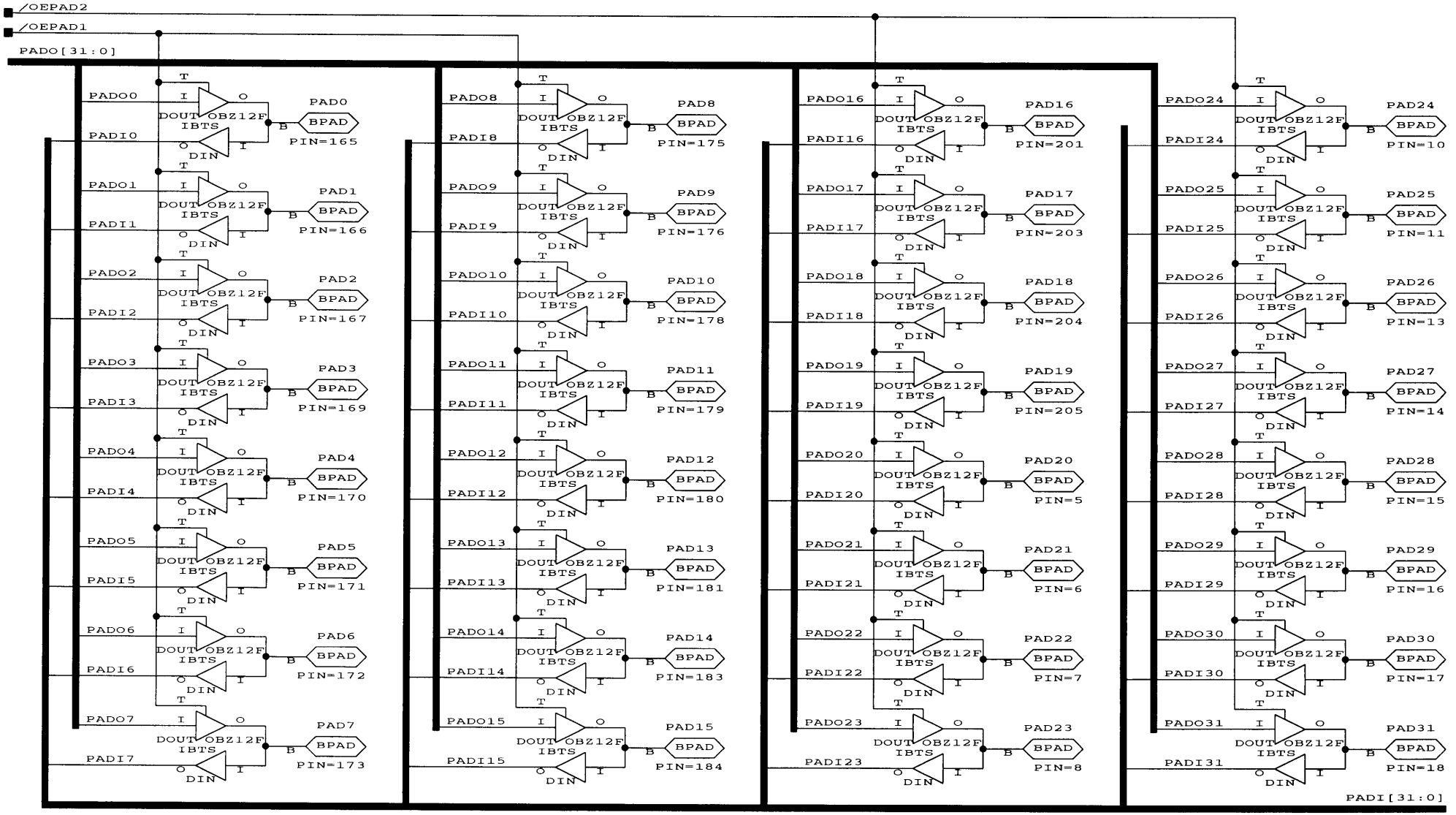



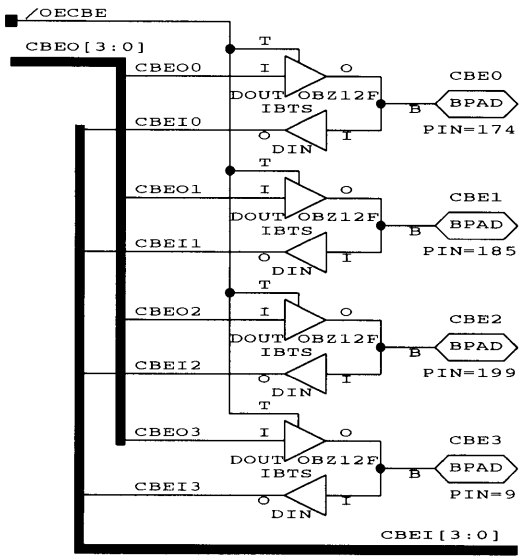



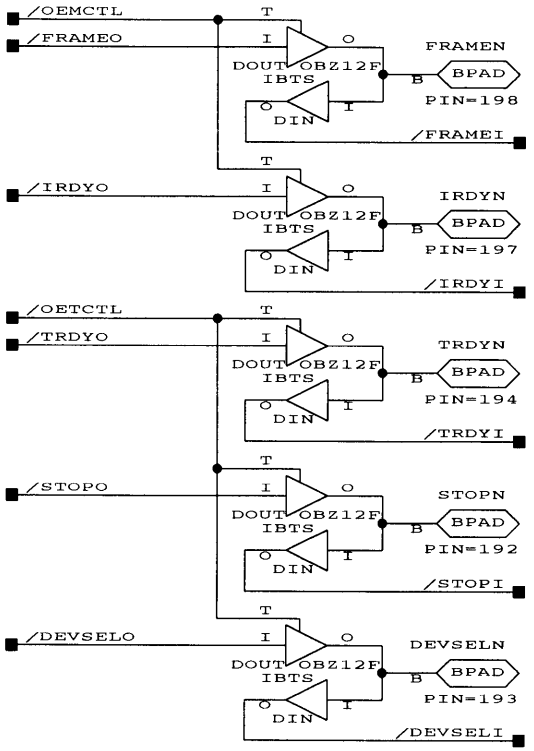



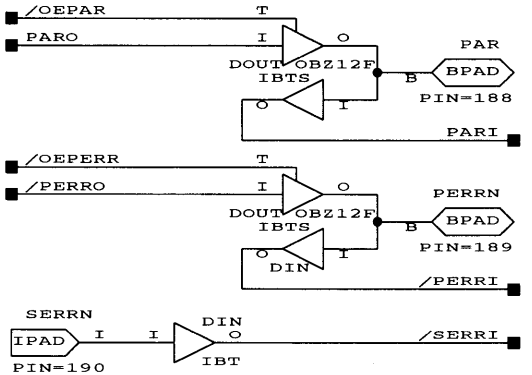

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>SMI CONTROL SIGNALS | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: bsp301 | Page: 3 of 6 |

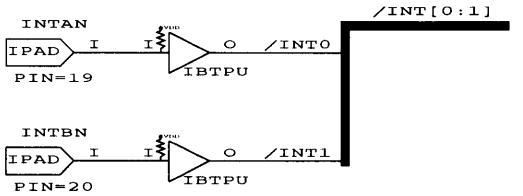


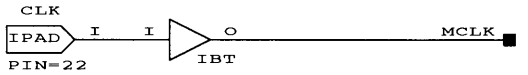

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 PCI ADDR/DATA TRANSCEIVERS | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: bsp301 | Page: 4 of 6 |

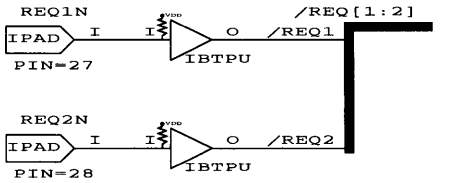


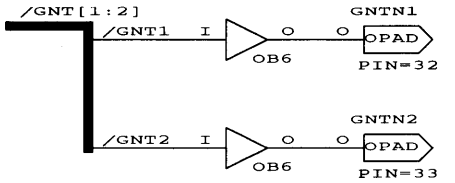



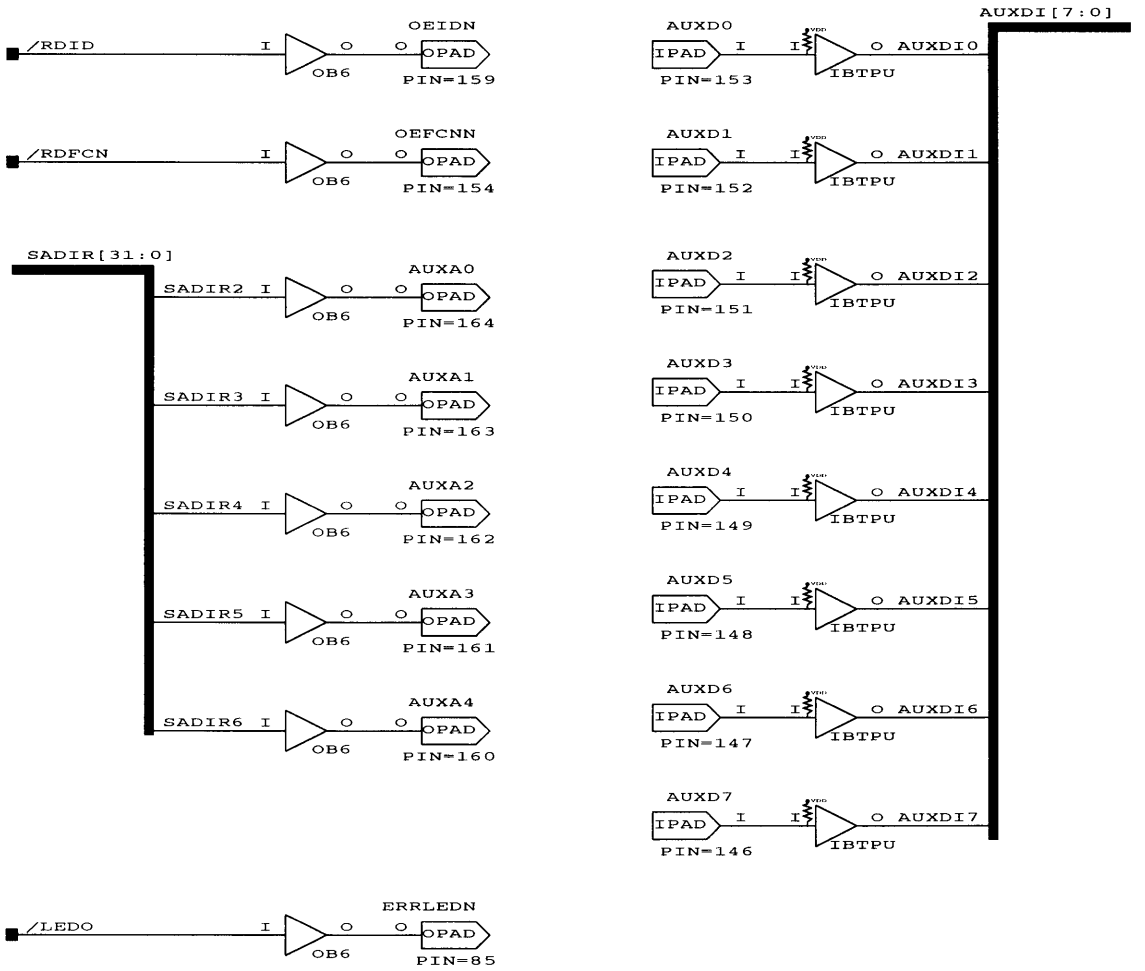



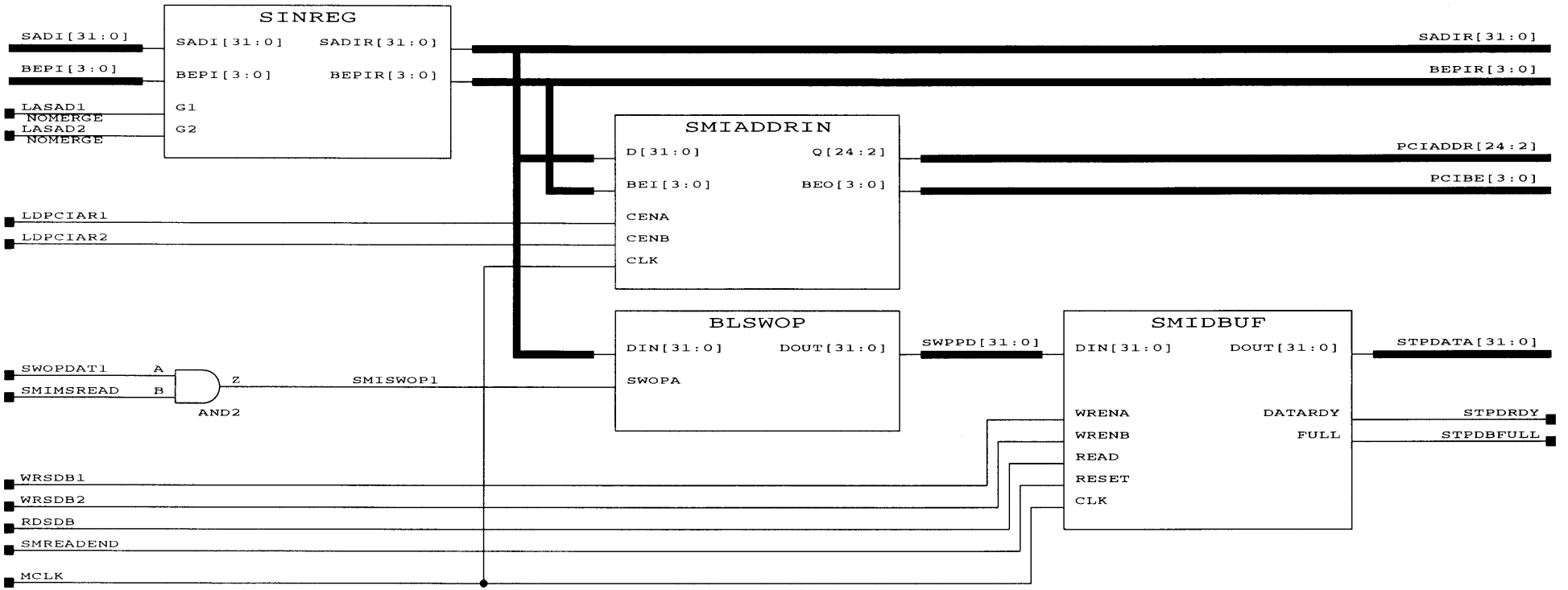



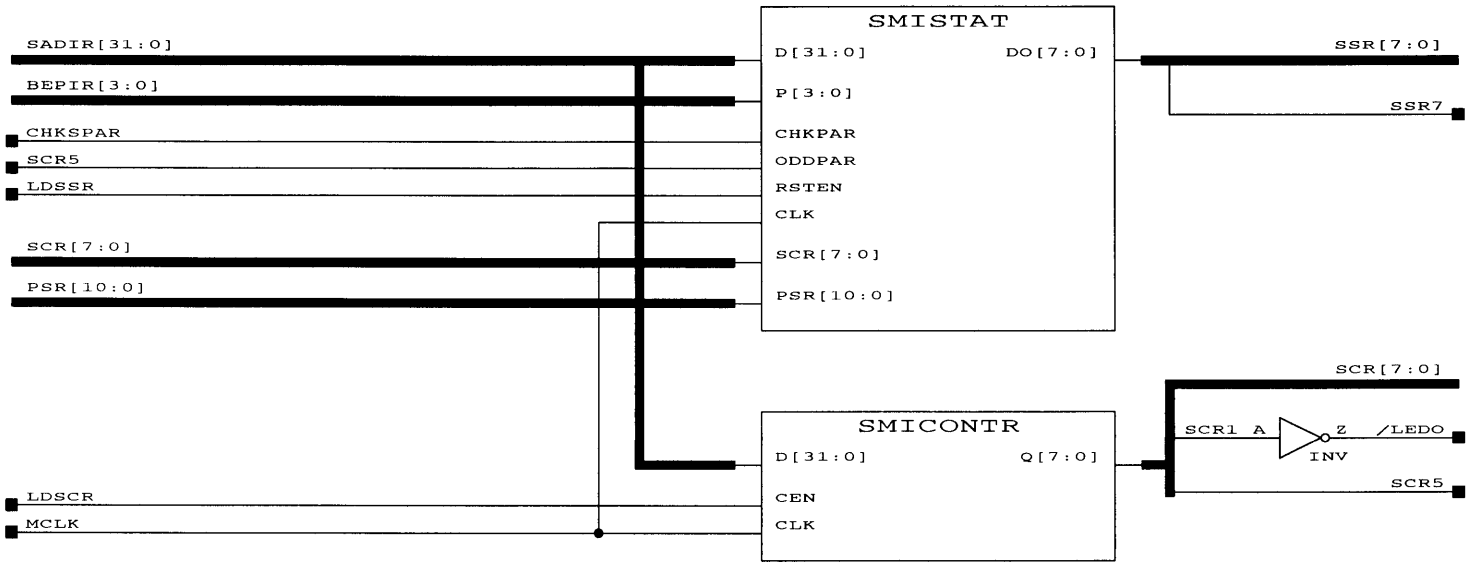

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970917 | BSP301-1 |
| Issue 1 | | PCI CONTROL SIGNALS |
| Issue 2 | | |
| Issue 3 | | File: bsp301 Page: 5 of 6 |



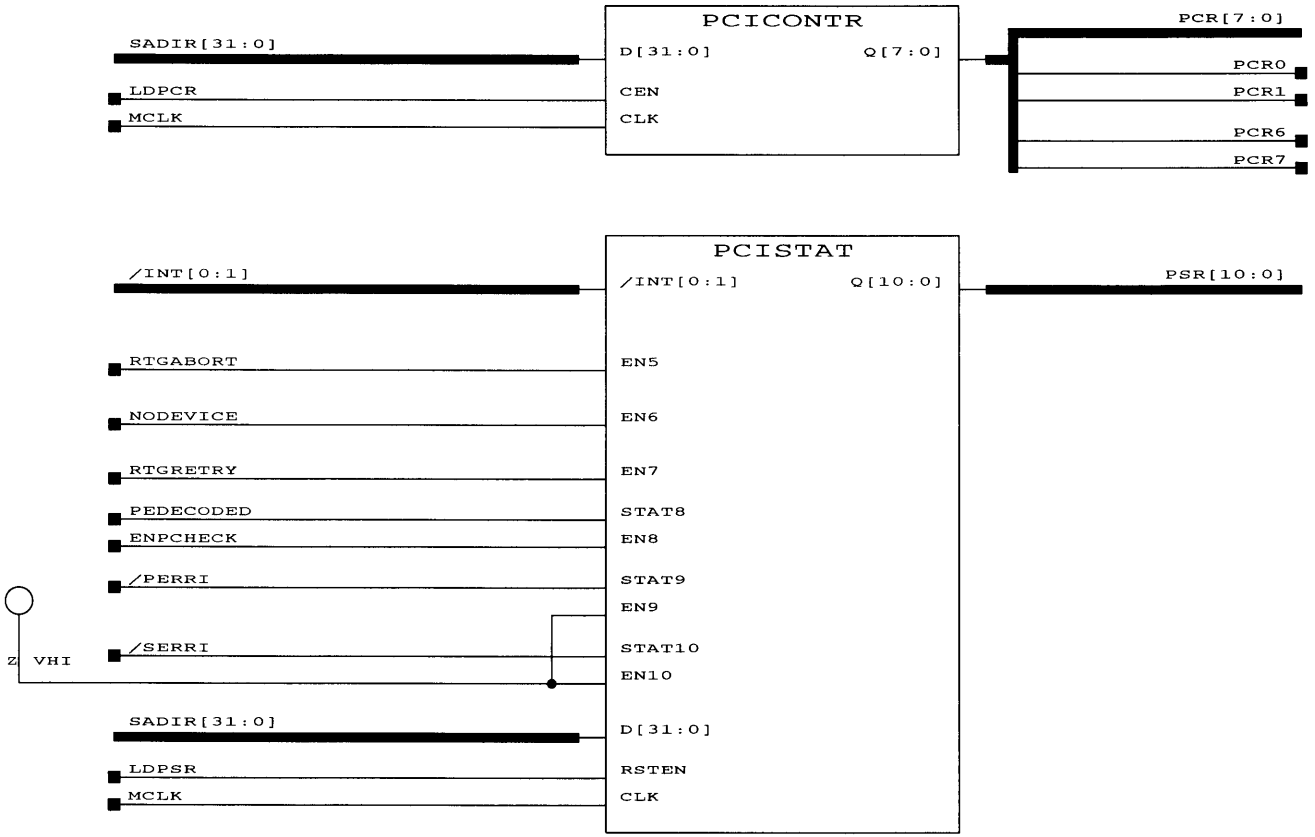

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>ID & FCN PROM INTERFACE |
| Issue 1 | | |
| Issue 2 | | |
| Issue 3 | | File: bsp301 Page: 6 of 6 |



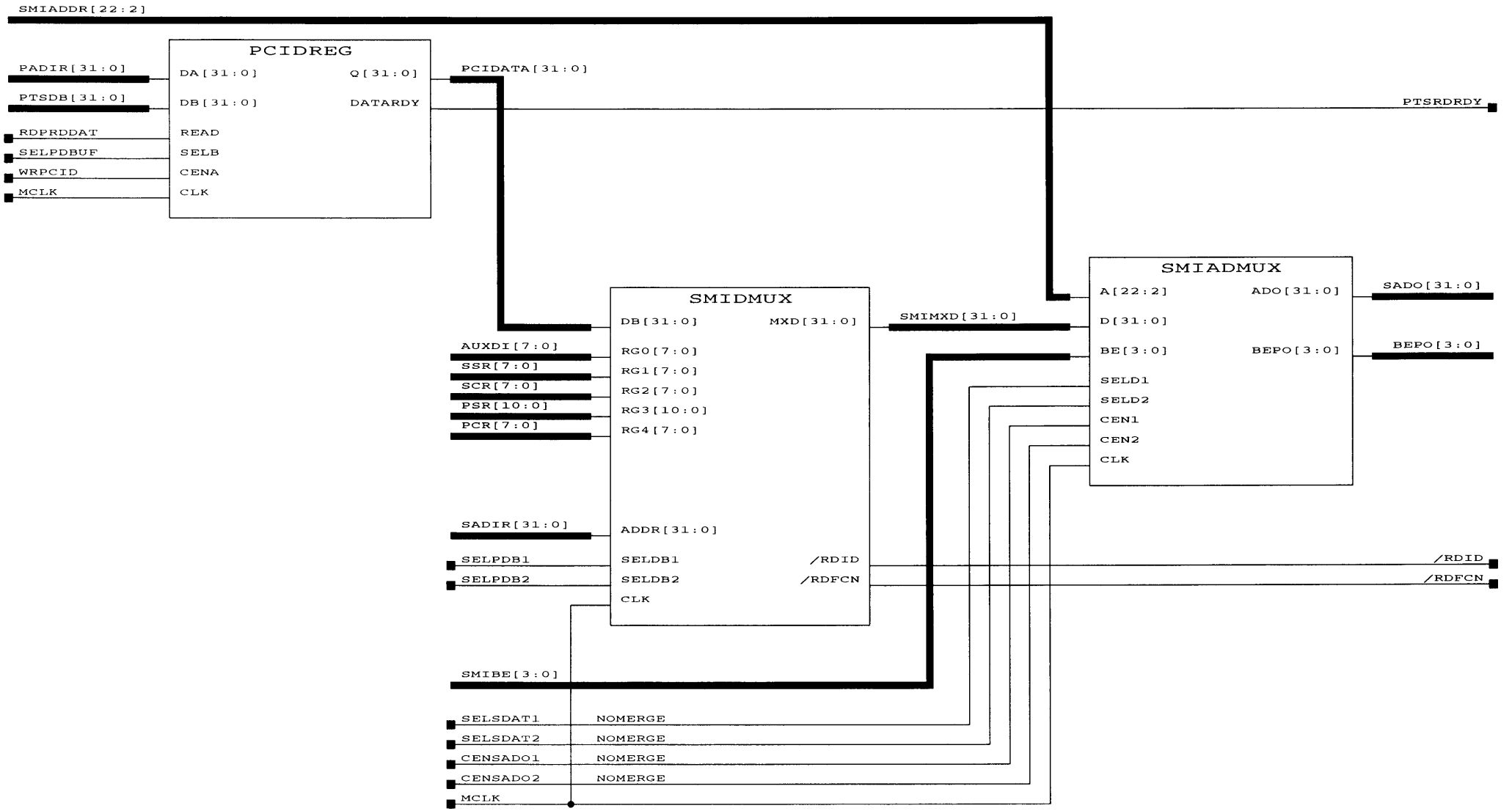

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: bspmain | Page: 1 of 8 |



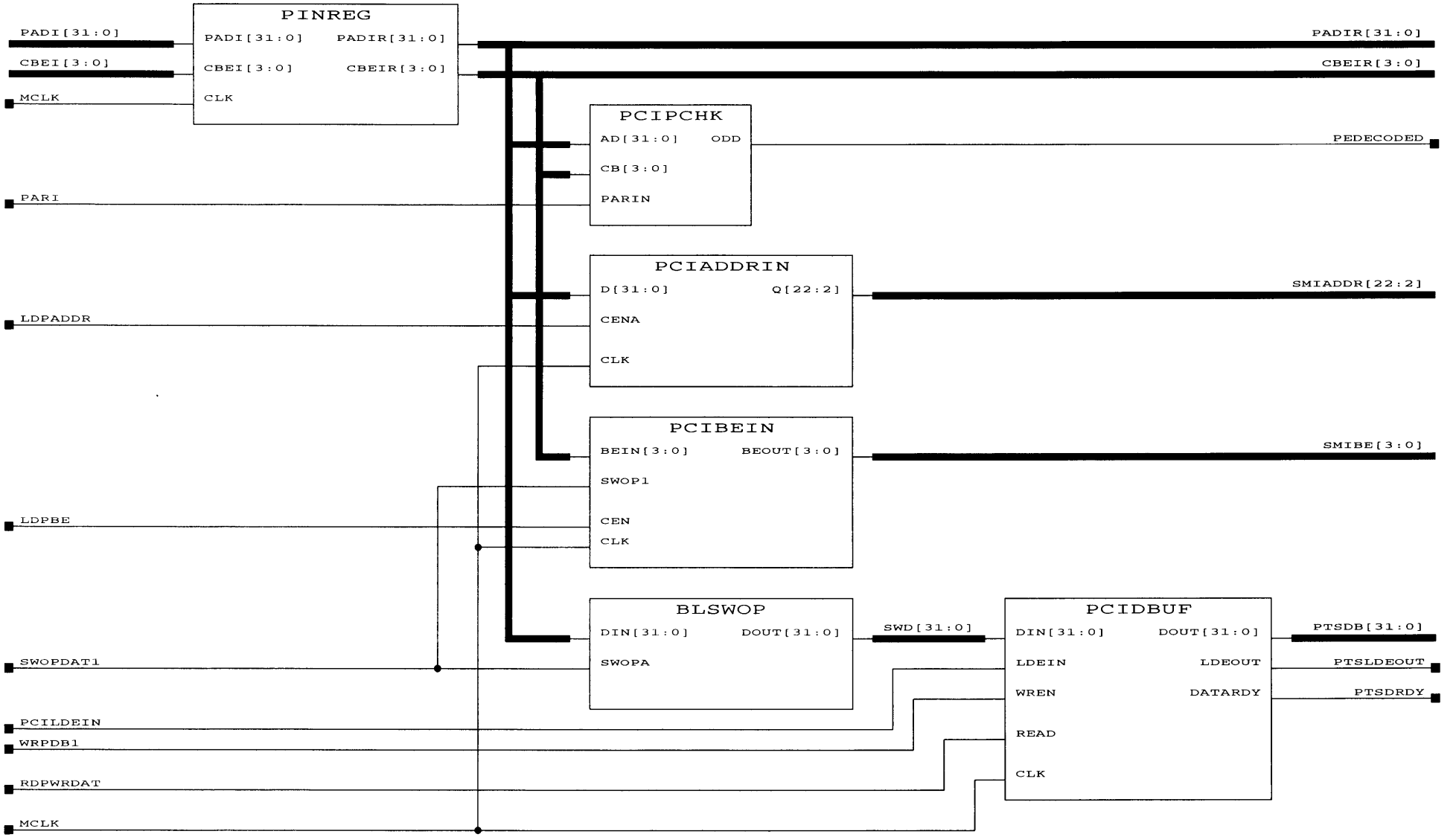

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: bspmain | Page:2 of 8 |



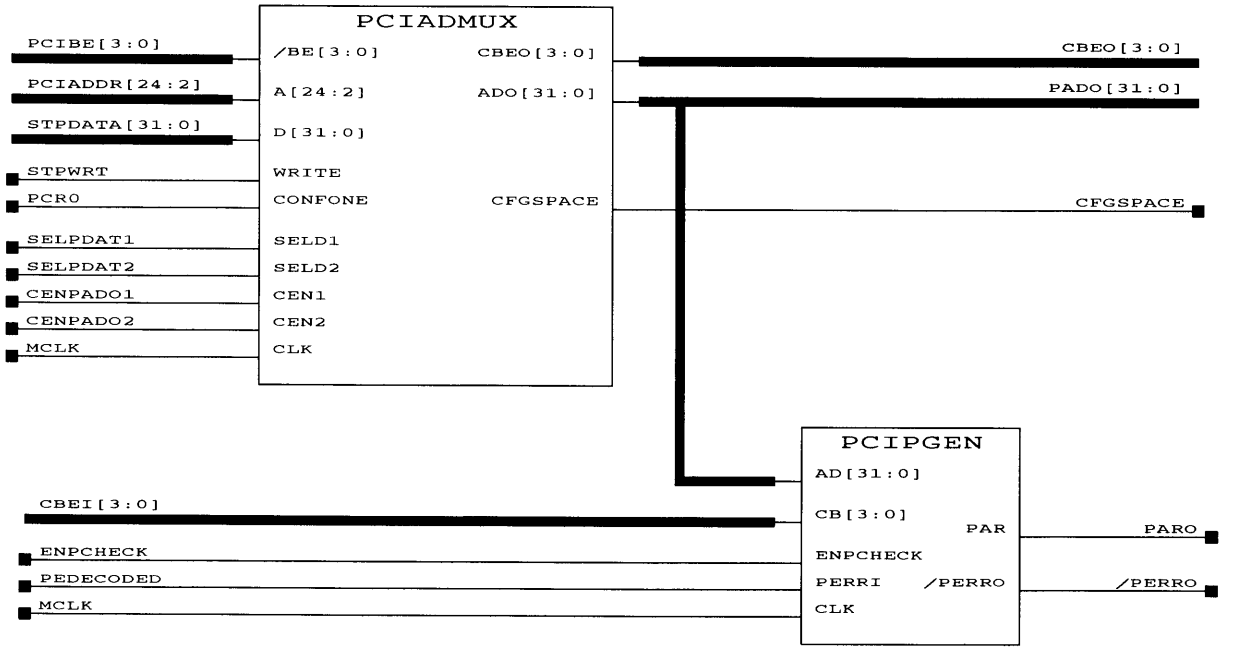

| | | | |
|---|---|---|---|
| dde | Dansk Data Elektronik A/S | | |
| Issue 0 | 970917 | BSP301-1 | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: bspmain | Page:3 of 8 |



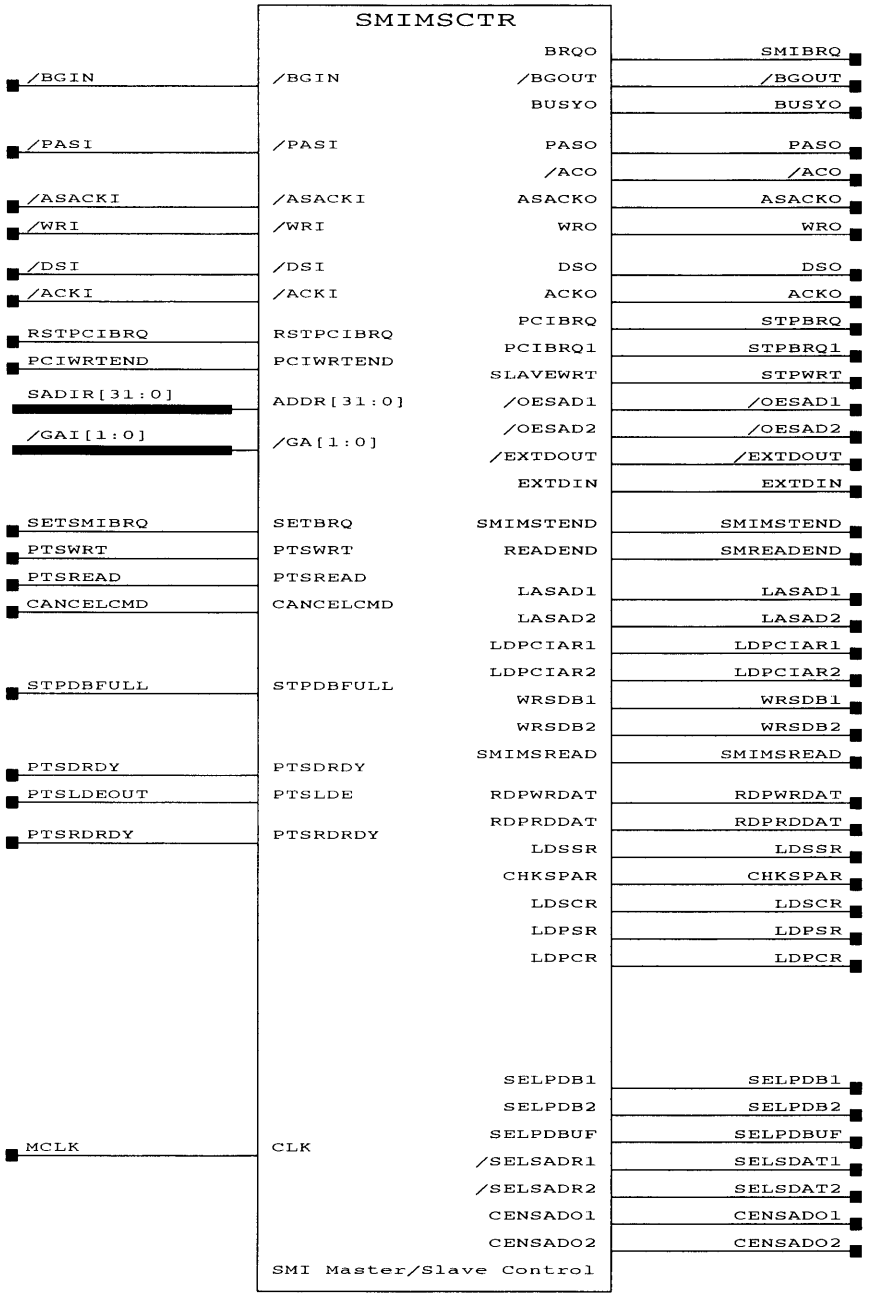

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: bspmain | Page:4 of 8 |



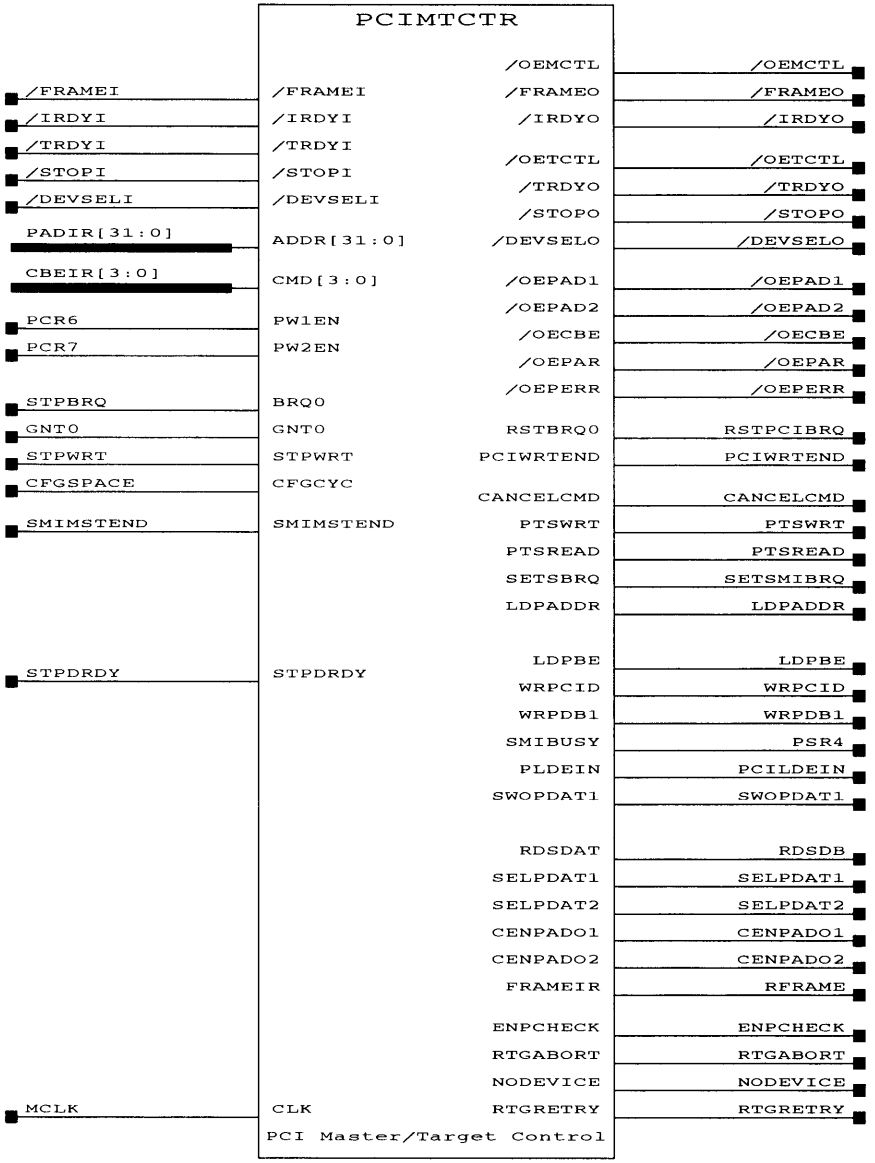

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | |
| Issue 2 | | |
| Issue 3 | | File: bspmain Page: 5 of 8 |

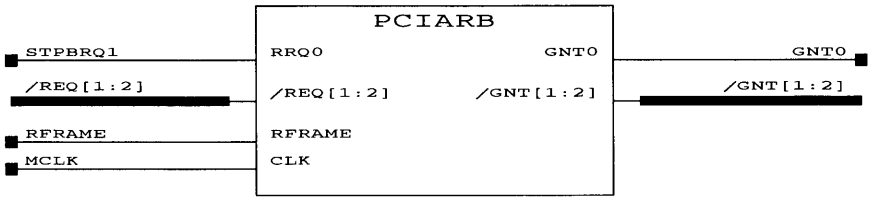



| | | | |
|---|---|---|---|
| dde | Dansk Data Elektronik A/S | | |
| Issue 0 | 970917 | BSP301-1 | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: bspmain | Page:6 of 8 |

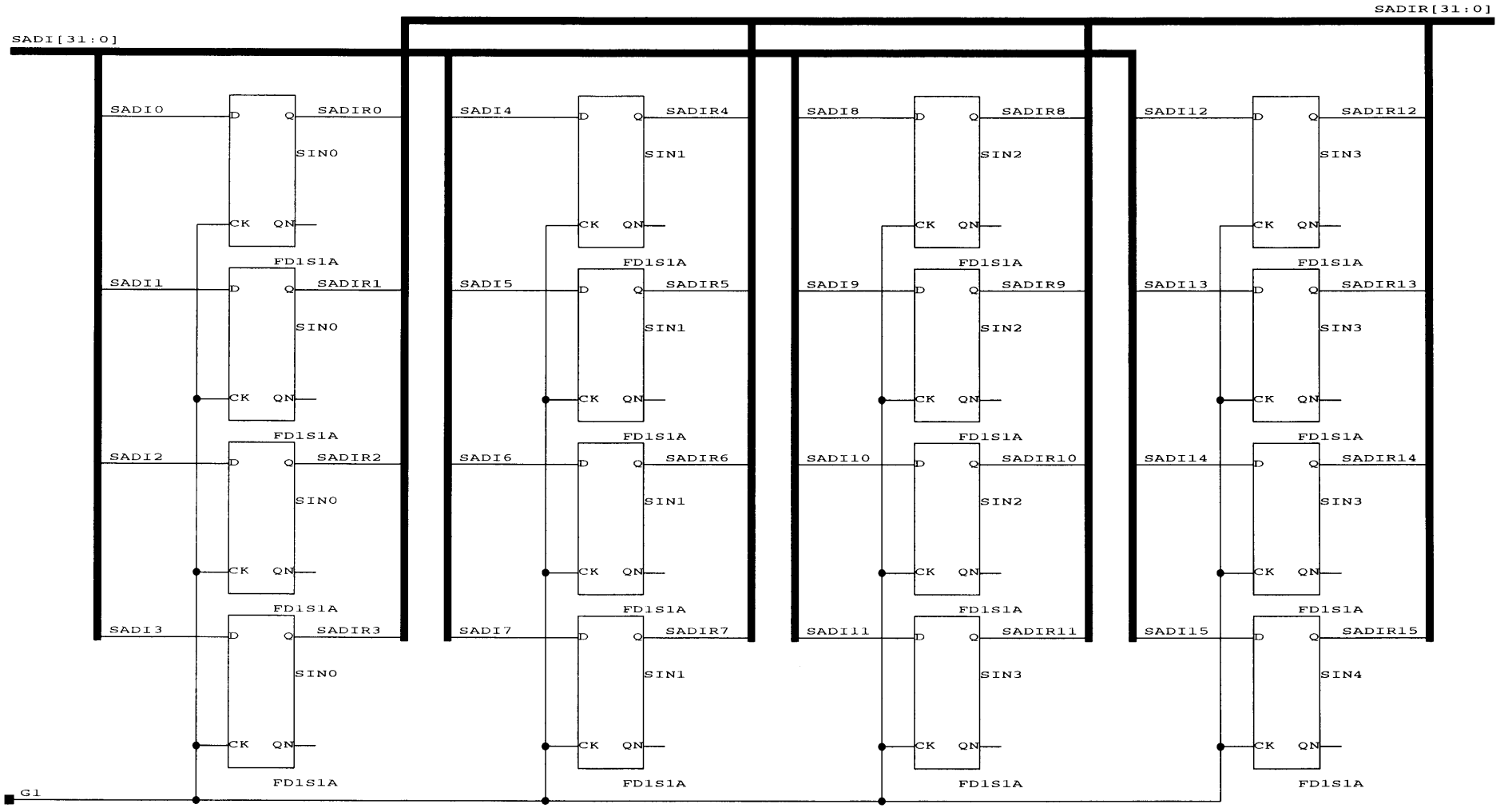



| | | | |
|---|---|---|---|
| dde | Dansk Data Elektronik A/S | | |
| Issue 0 | 970901 | BSP301-1 | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: bspmain | Page:7 of 8 |

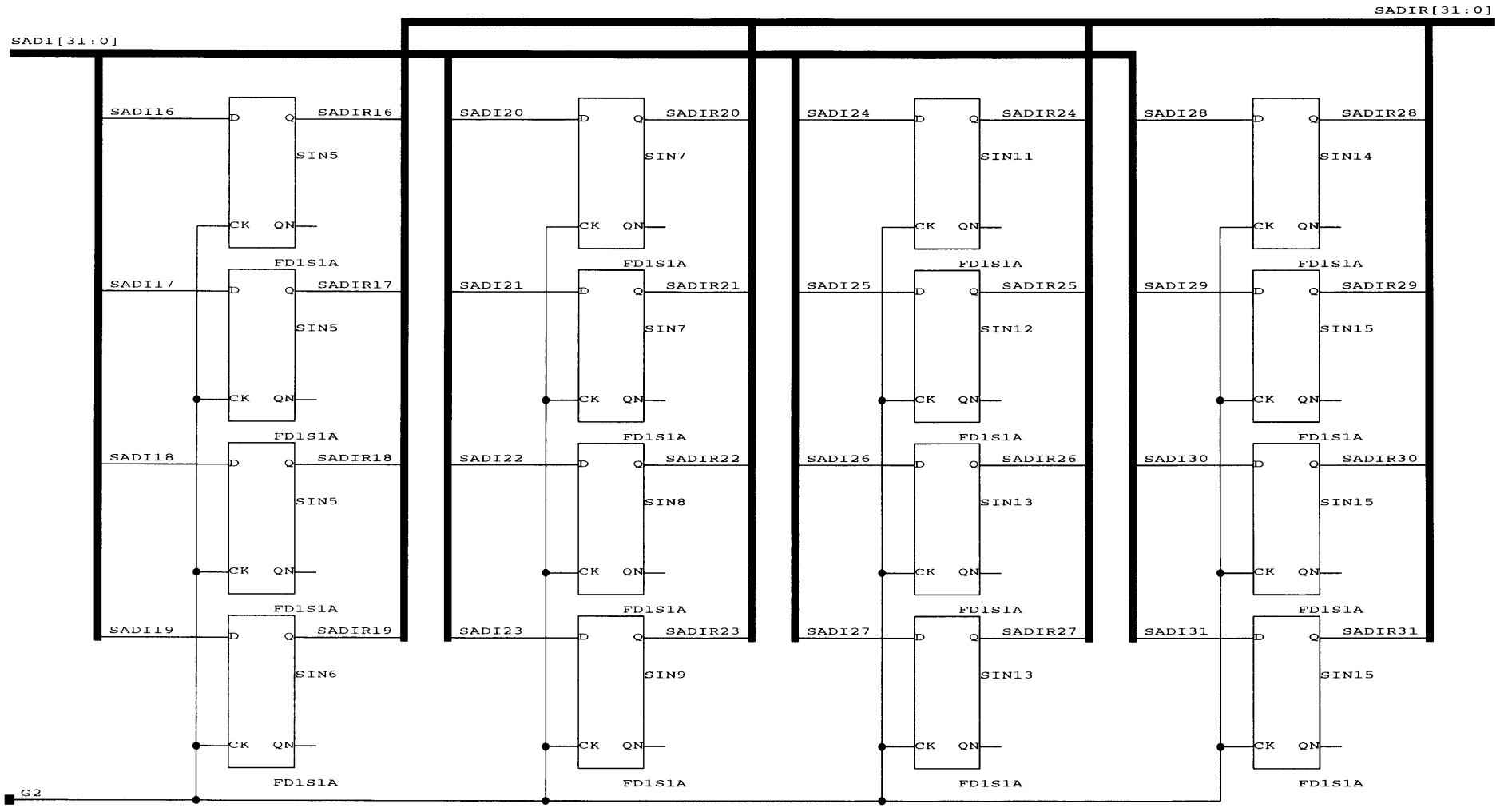


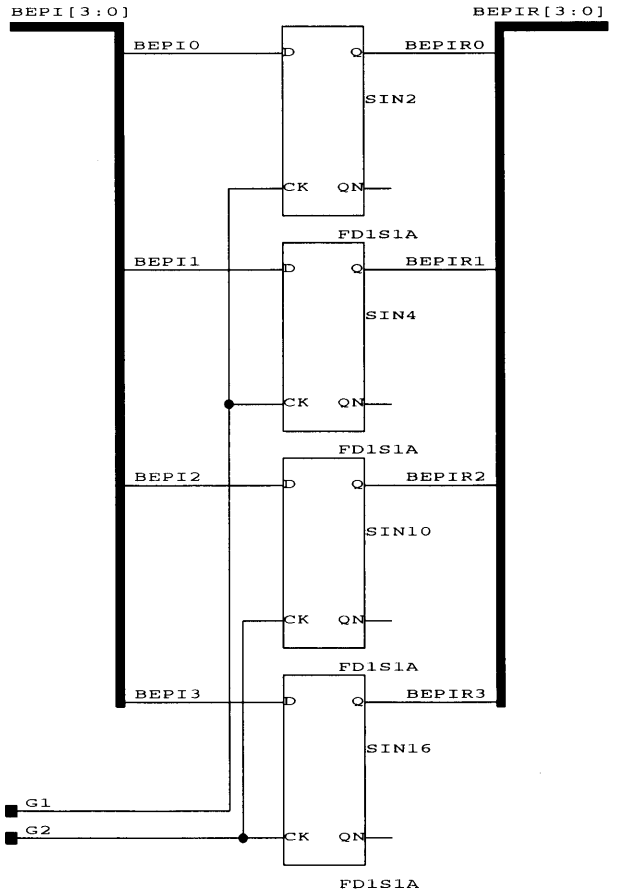

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970917 | BSP301-1 |
| Issue 1 | | |
| Issue 2 | | |
| Issue 3 | | File: bspmain Page:8 of 8 |



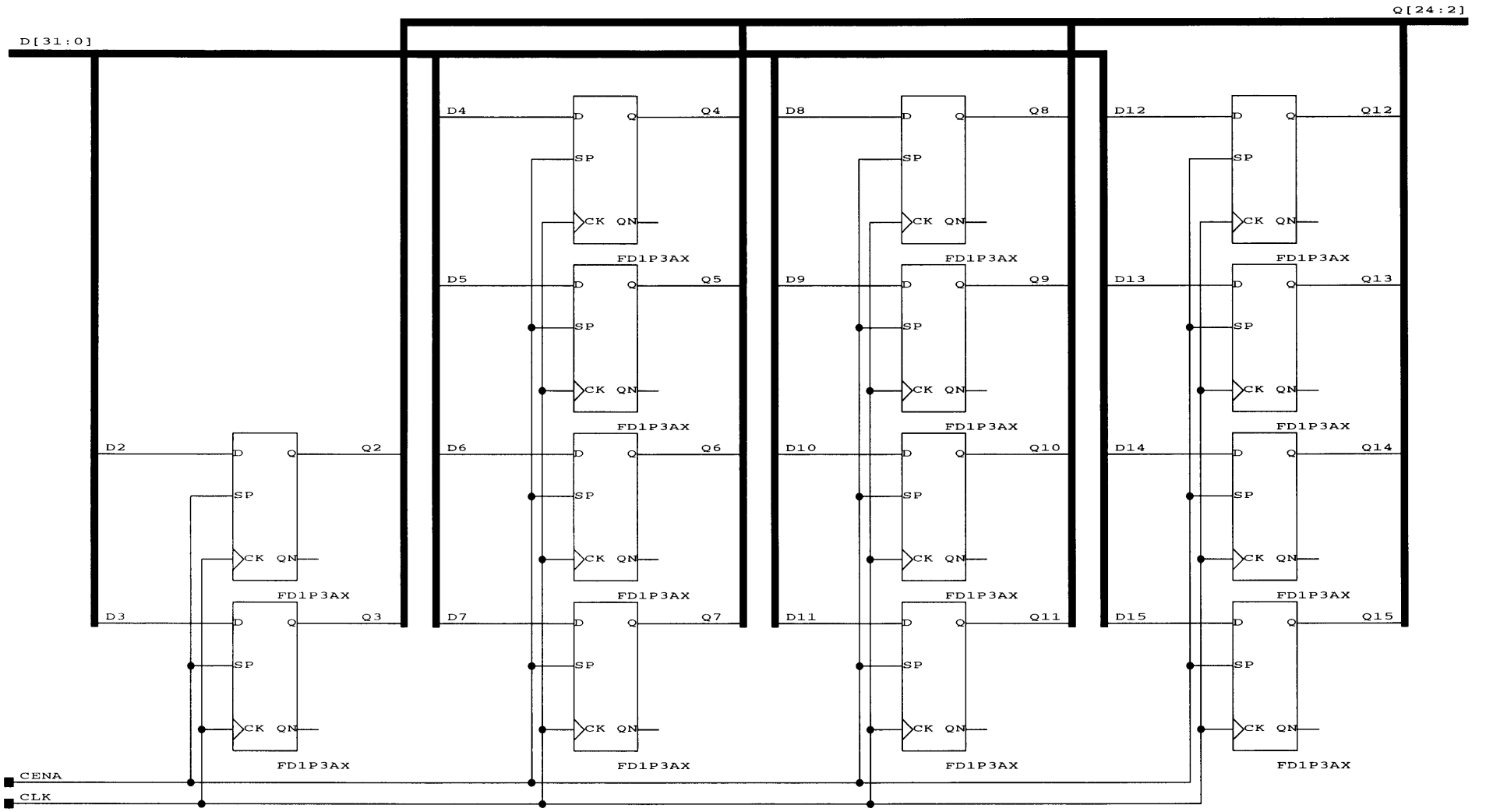

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI INPUT REGISTER |
| Issue 2 | | |
| Issue 3 | | File: sinreg Page:1 of 3 |



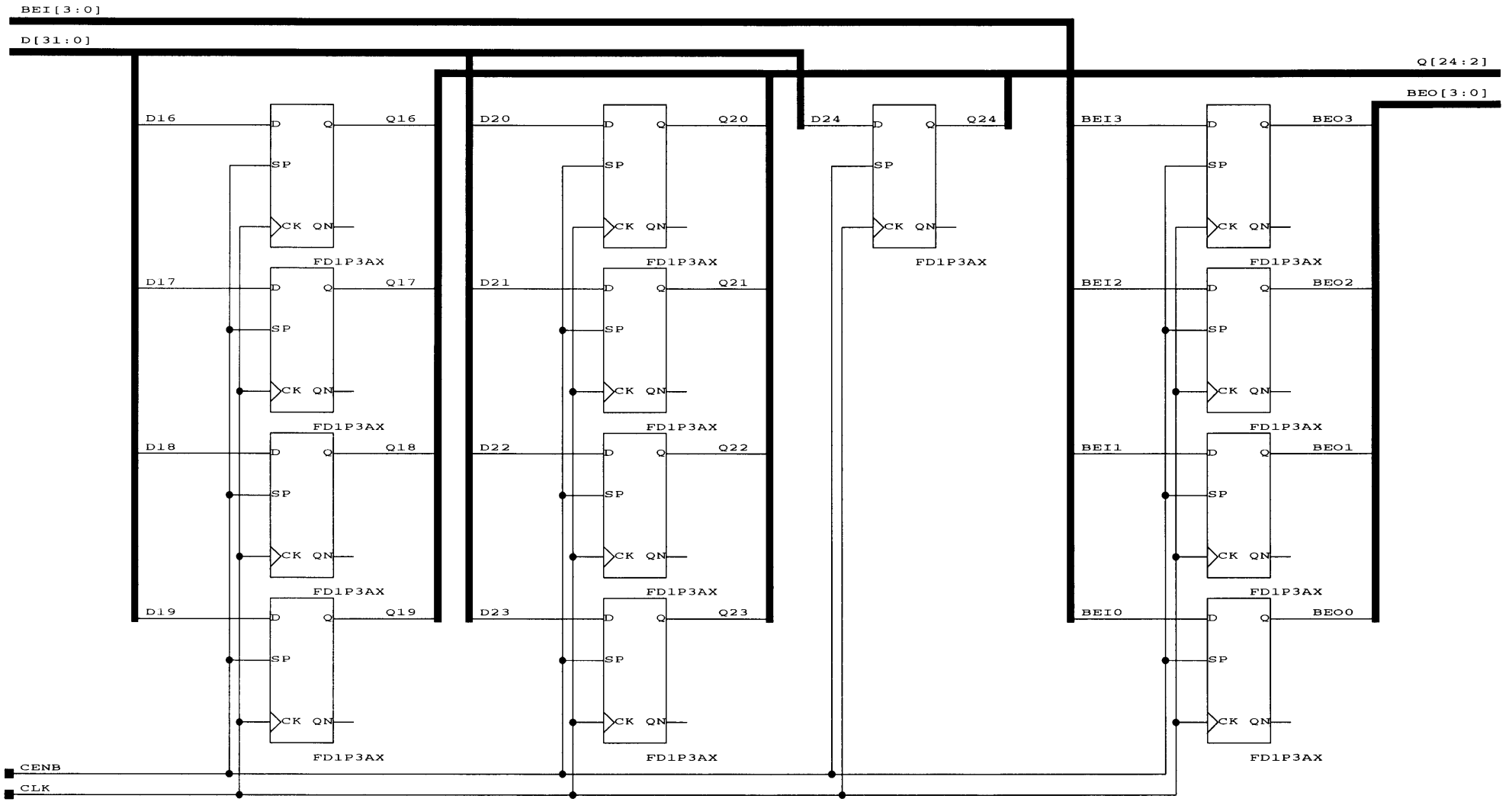

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI INPUT REGISTER |
| Issue 2 | | |
| Issue 3 | | File: sinreg Page:2 of 3 |



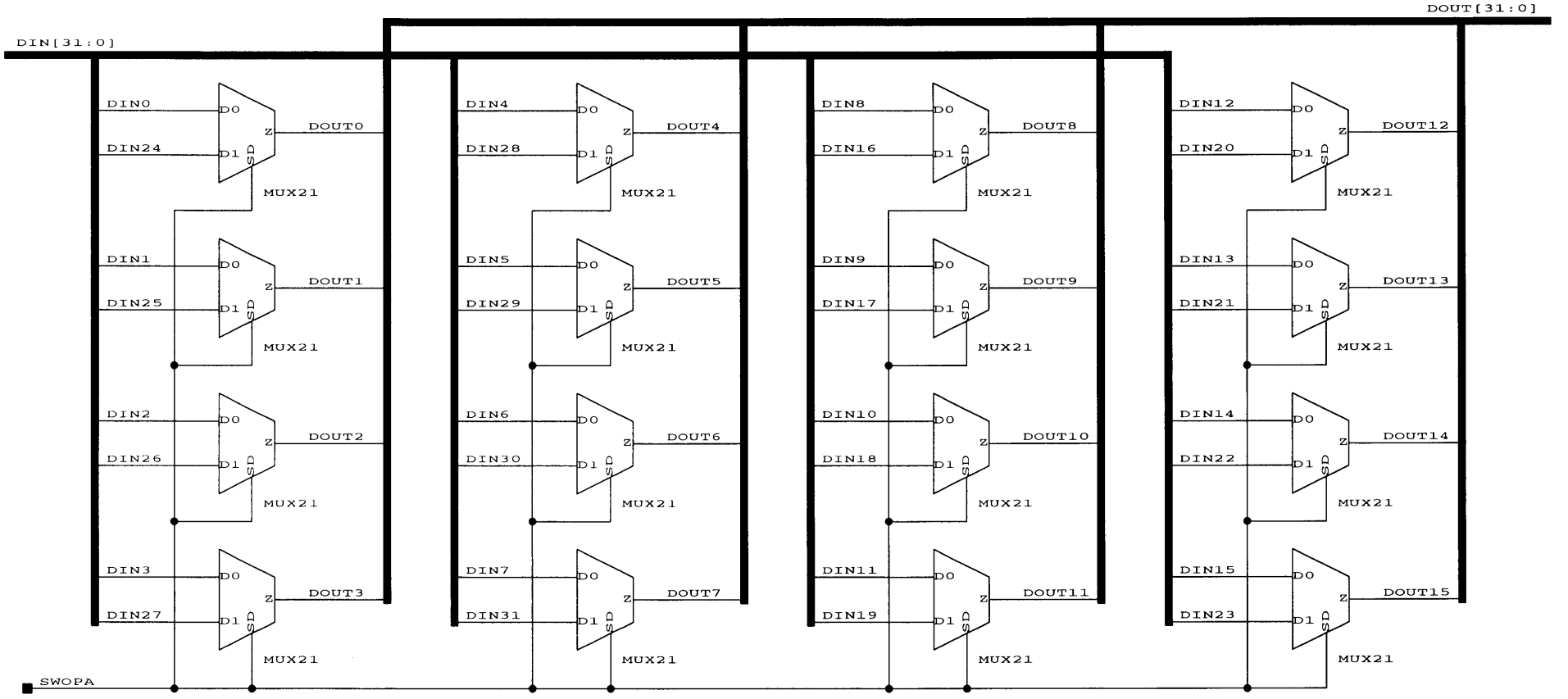

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>SMI INPUT REGISTER |
| Issue 1 | | |
| Issue 2 | | |
| Issue 3 | | File: sinreg Page:3 of 3 |



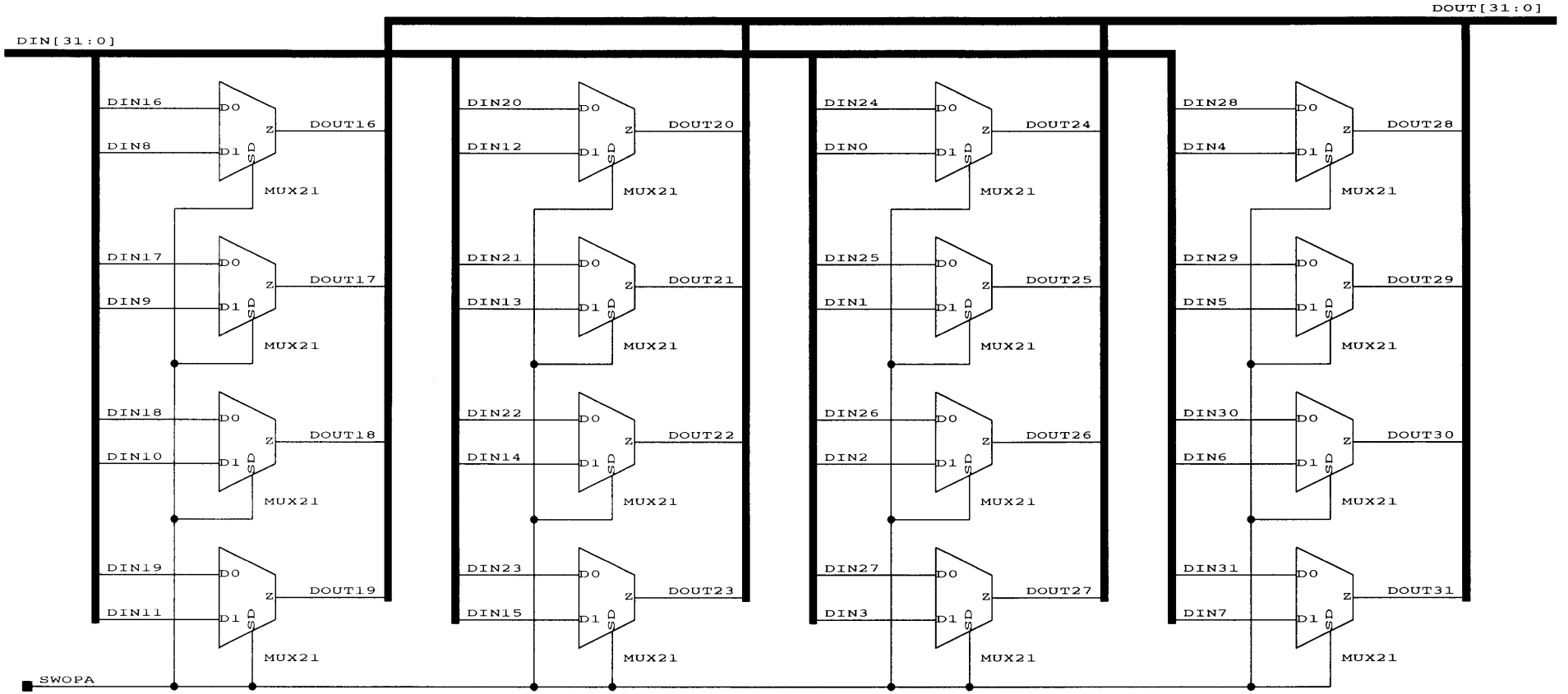

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI ADDRESS INPUT REGISTER |
| Issue 2 | | |
| Issue 3 | | File: smiaddrin Page:1 of 2 |



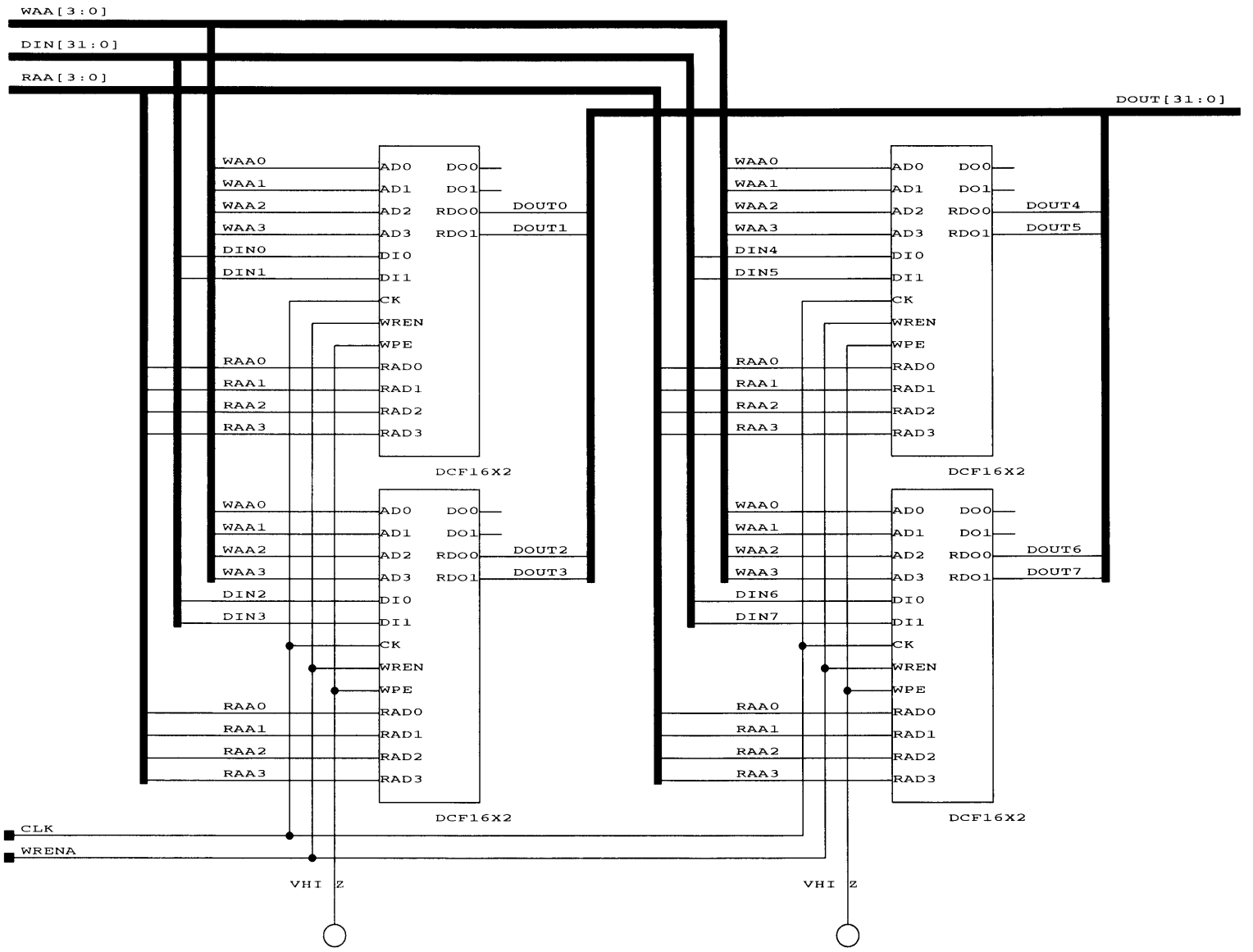

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI ADDRESS INPUT REGISTER |
| Issue 2 | | |
| Issue 3 | | File: smiaddrin Page:2 of 2 |



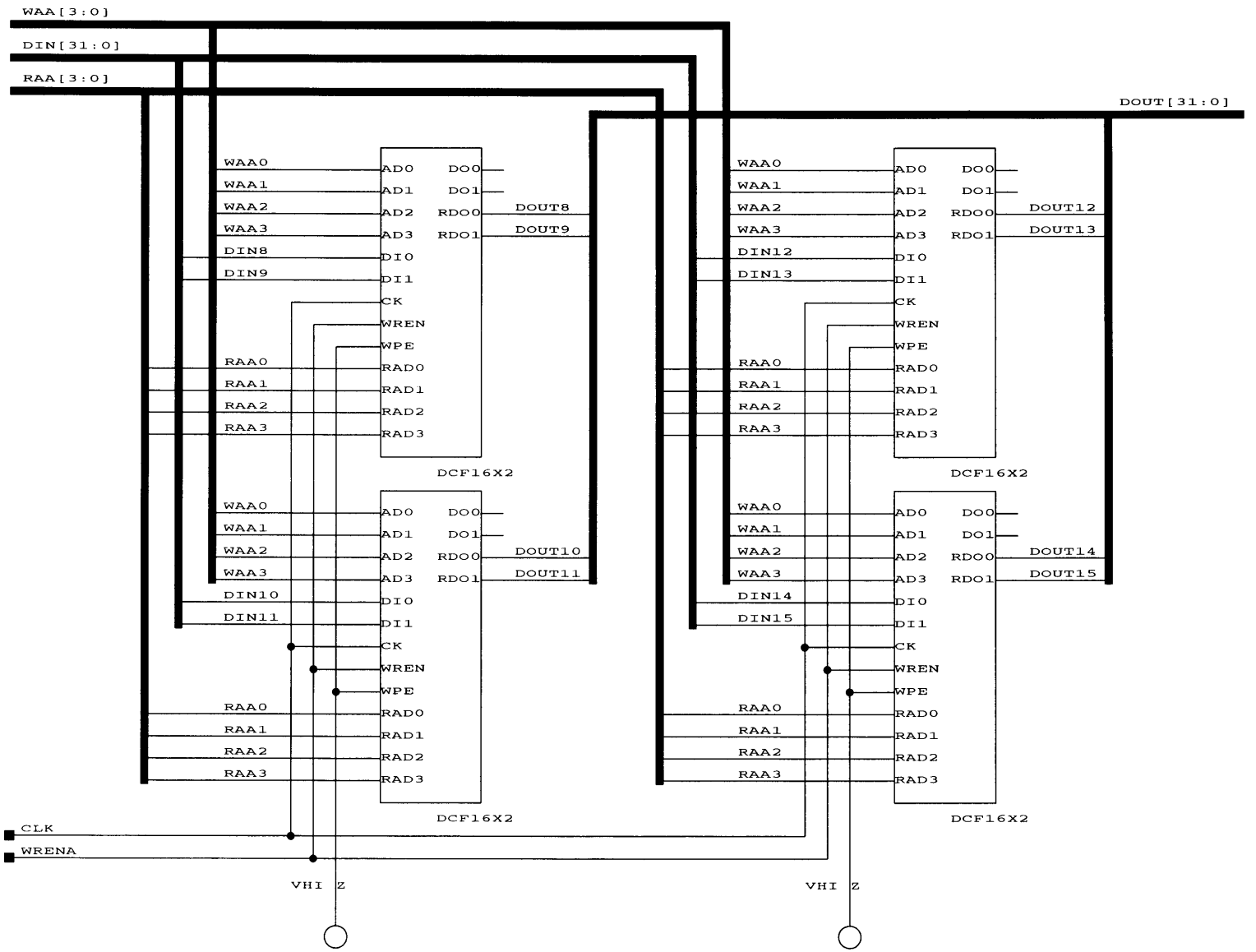

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>BYTE LANE SWOPPER | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: blswop | Page:1 of 2 |



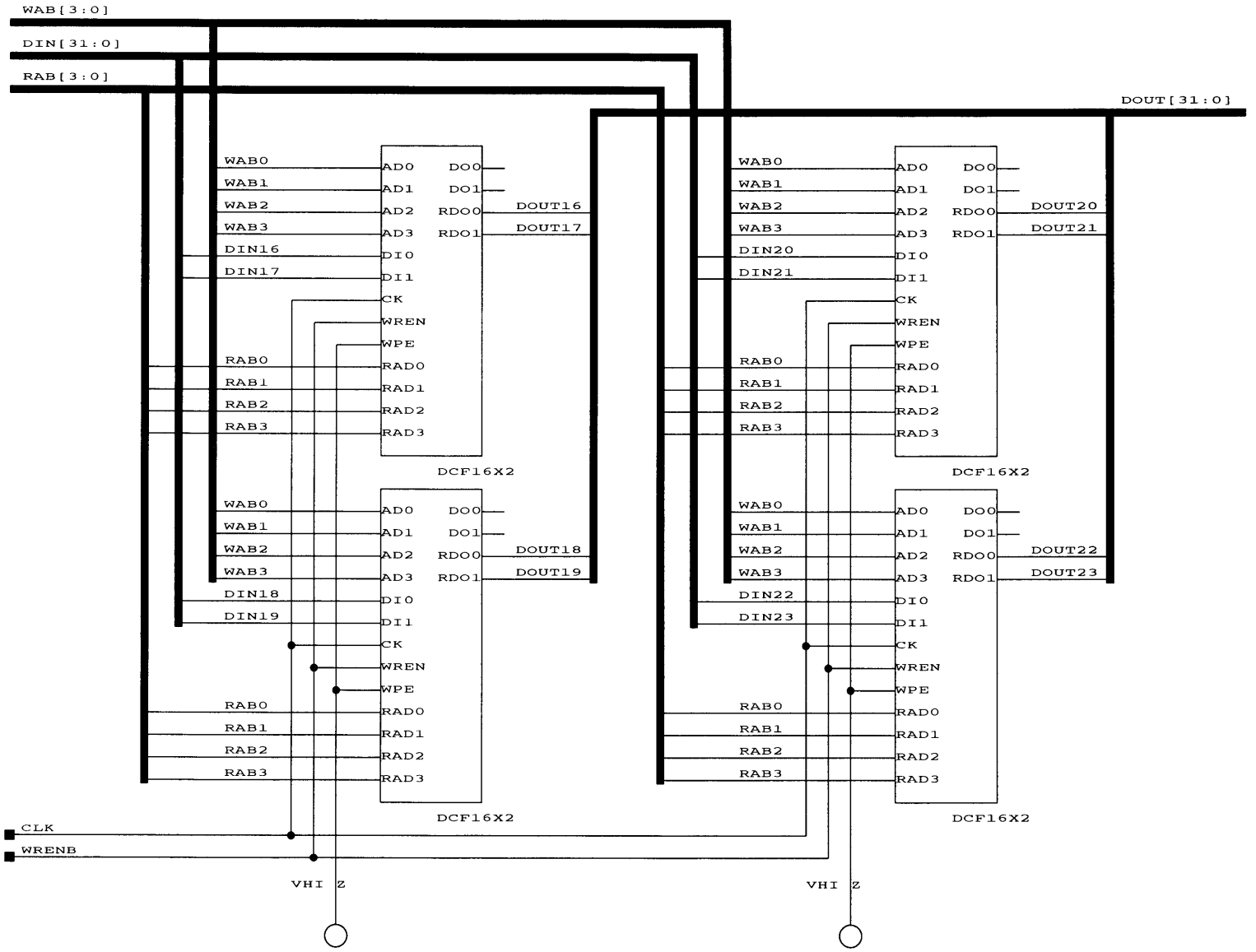

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>BYTE LANE SWOPPER | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: blswop | Page:2 of 2 |



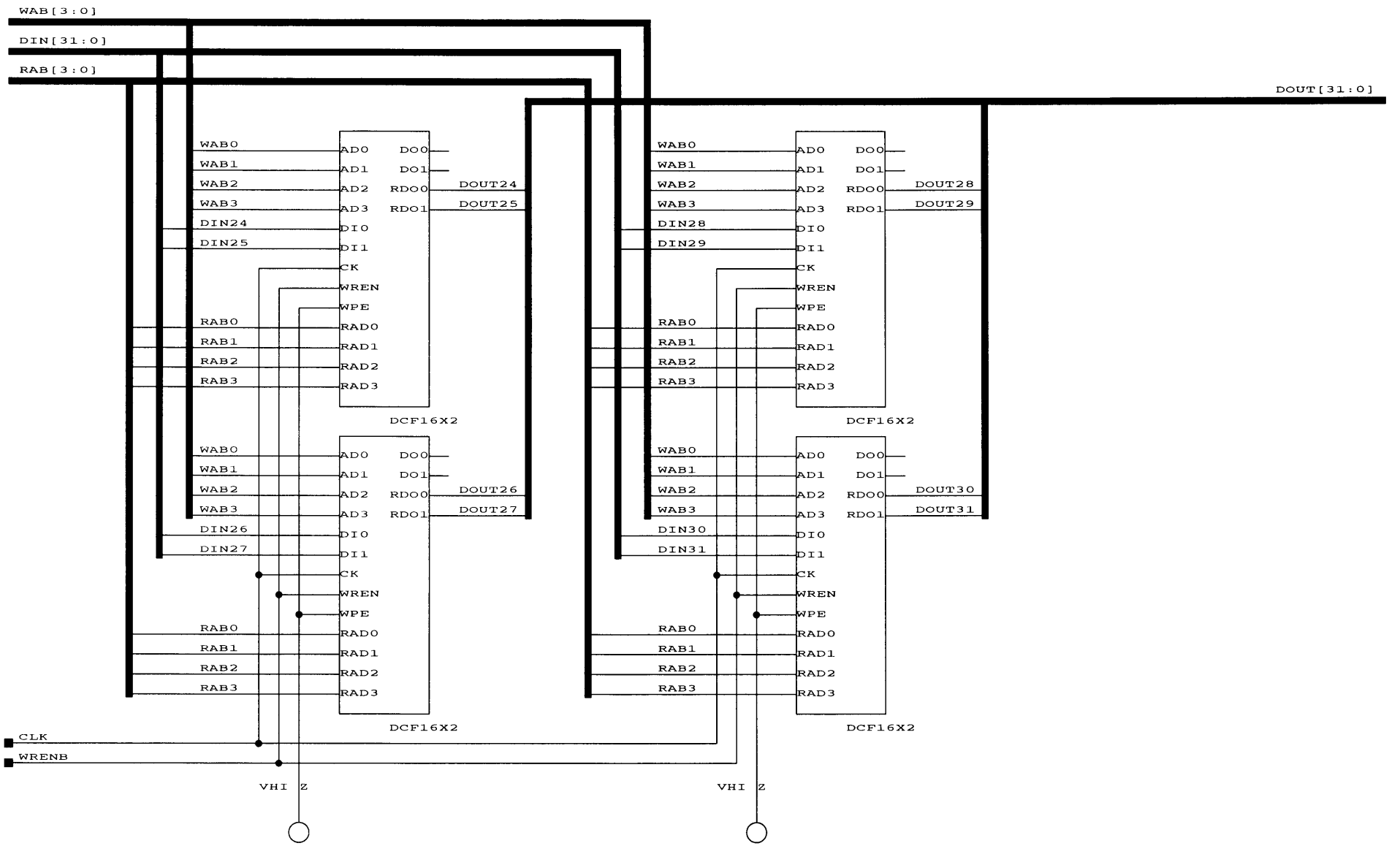

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 | |
| Issue 1 | | DATA BUFFER - 16 WORDS | |
| Issue 2 | | | |
| Issue 3 | | File: smidbuf | Page:1 of 5 |



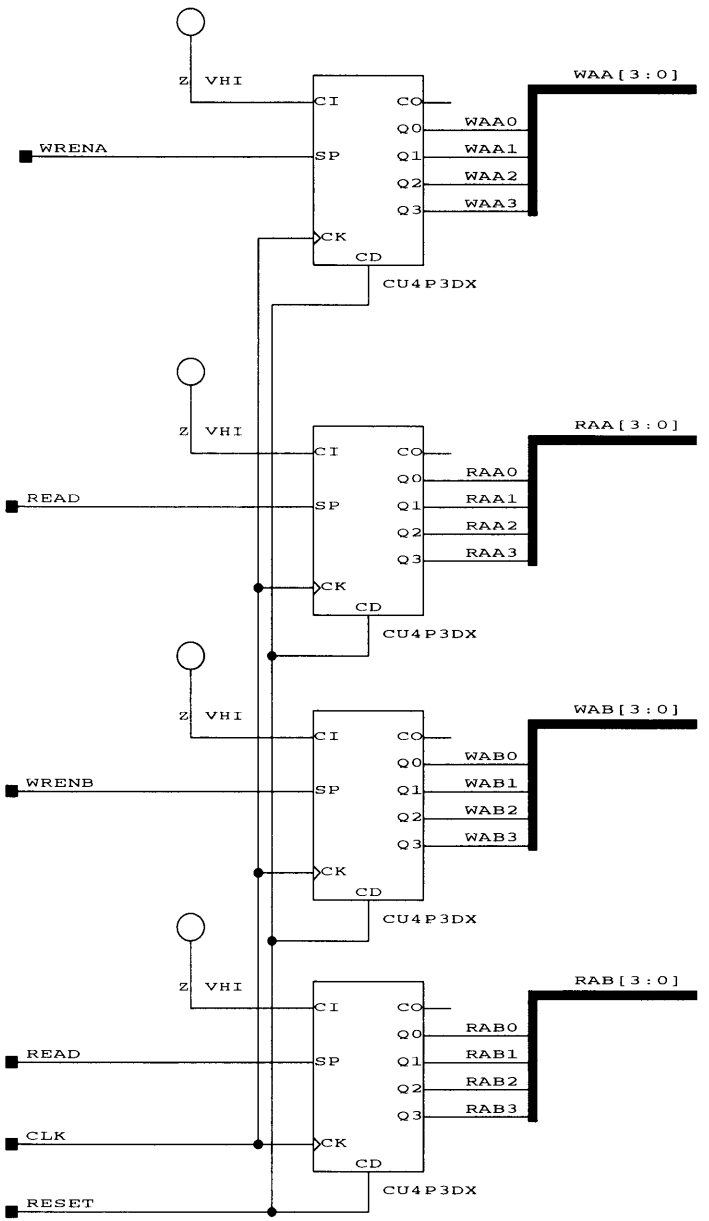

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>DATA BUFFER - 16 WORDS | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: smidbuf | Page:2 of 5 |

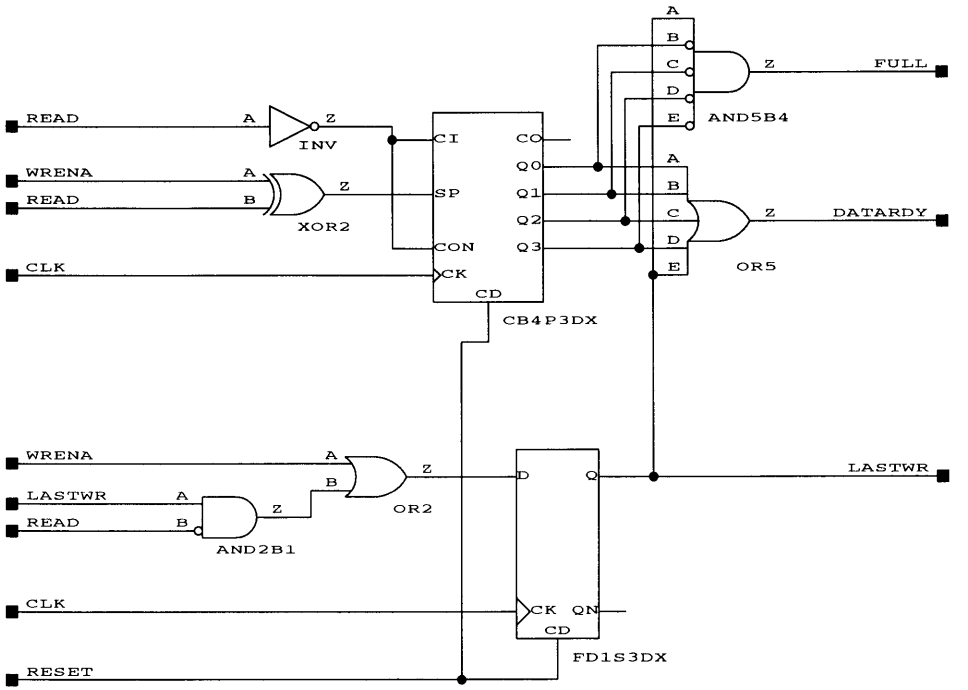



| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | DATA BUFFER - 16 WORDS |
| Issue 2 | | |
| Issue 3 | | File: smidbuf Page:3 of 5 |

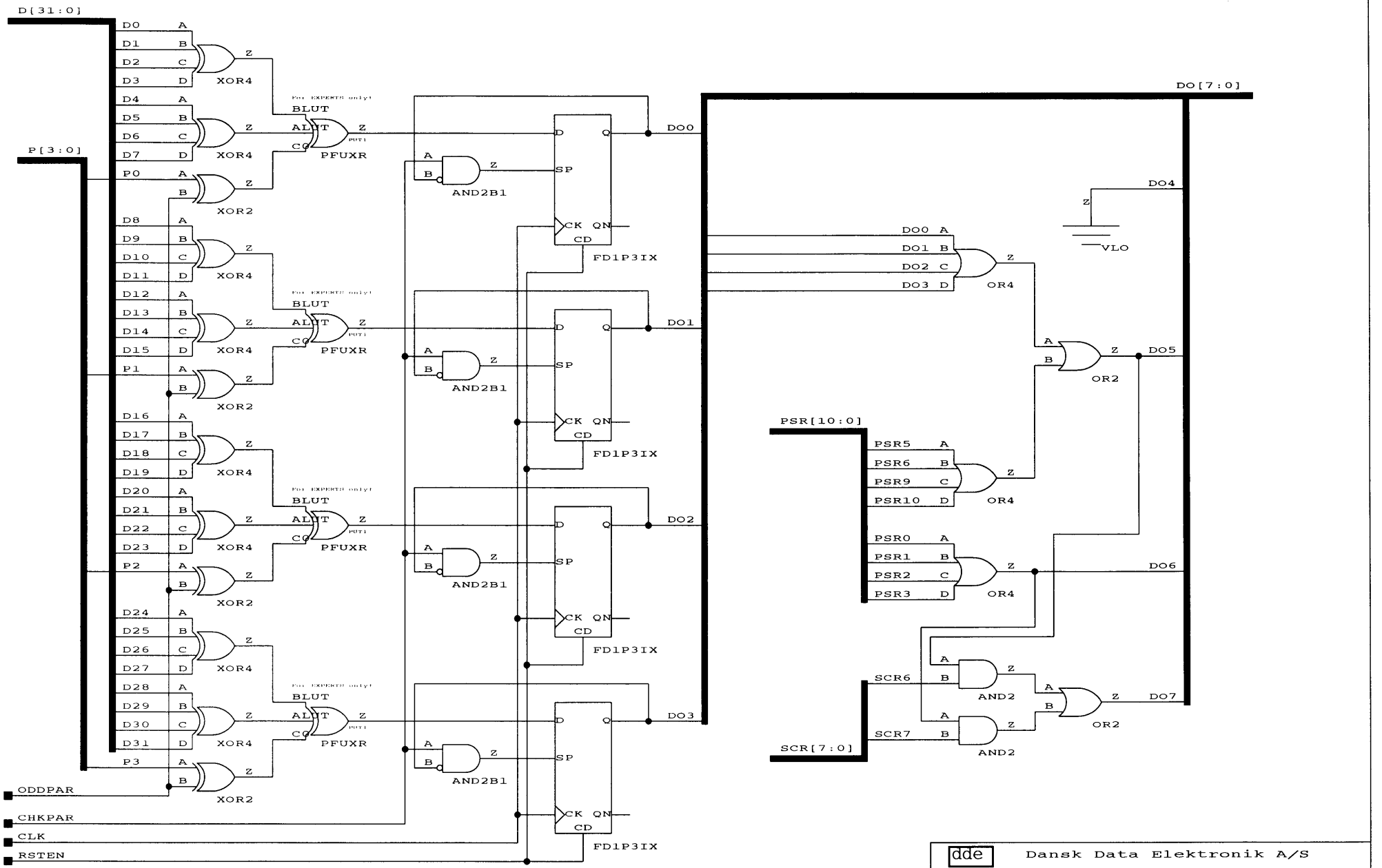



| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | DATA BUFFER - 16 WORDS |
| Issue 2 | | |
| Issue 3 | | File: smidbuf Page:4 of 5 |

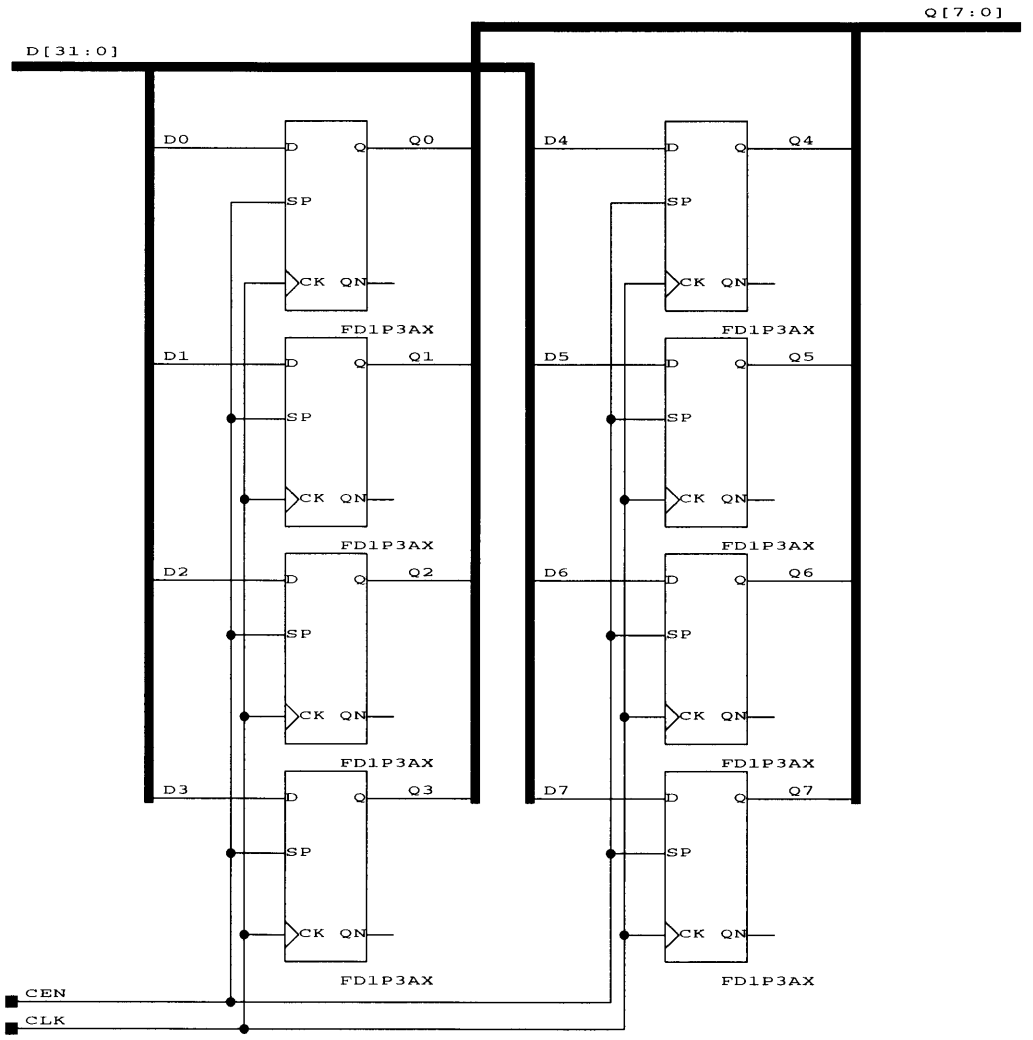


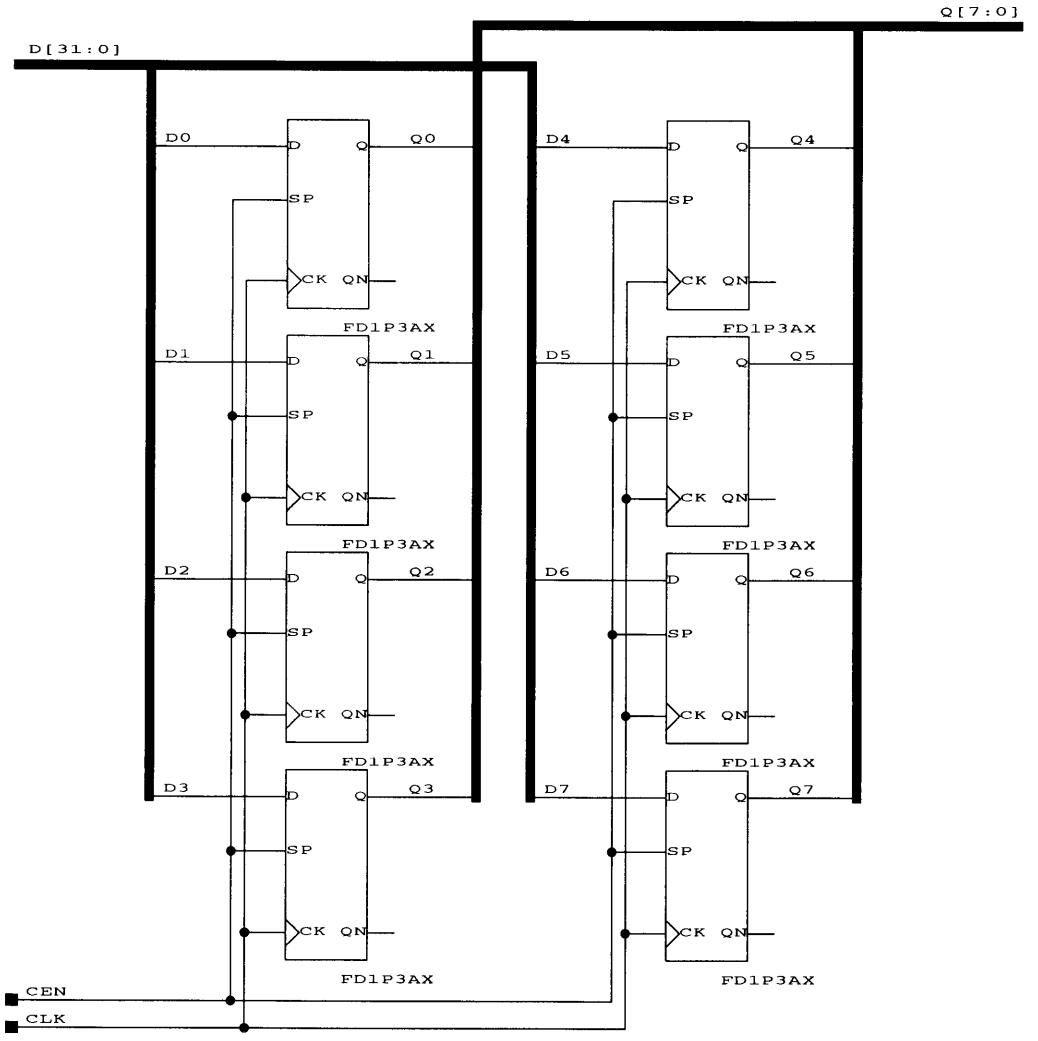

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>DATA BUFFER - 16 WORDS |
| Issue 1 | | |
| Issue 2 | | |
| Issue 3 | | File: smidbuf Page:5 of 5 |



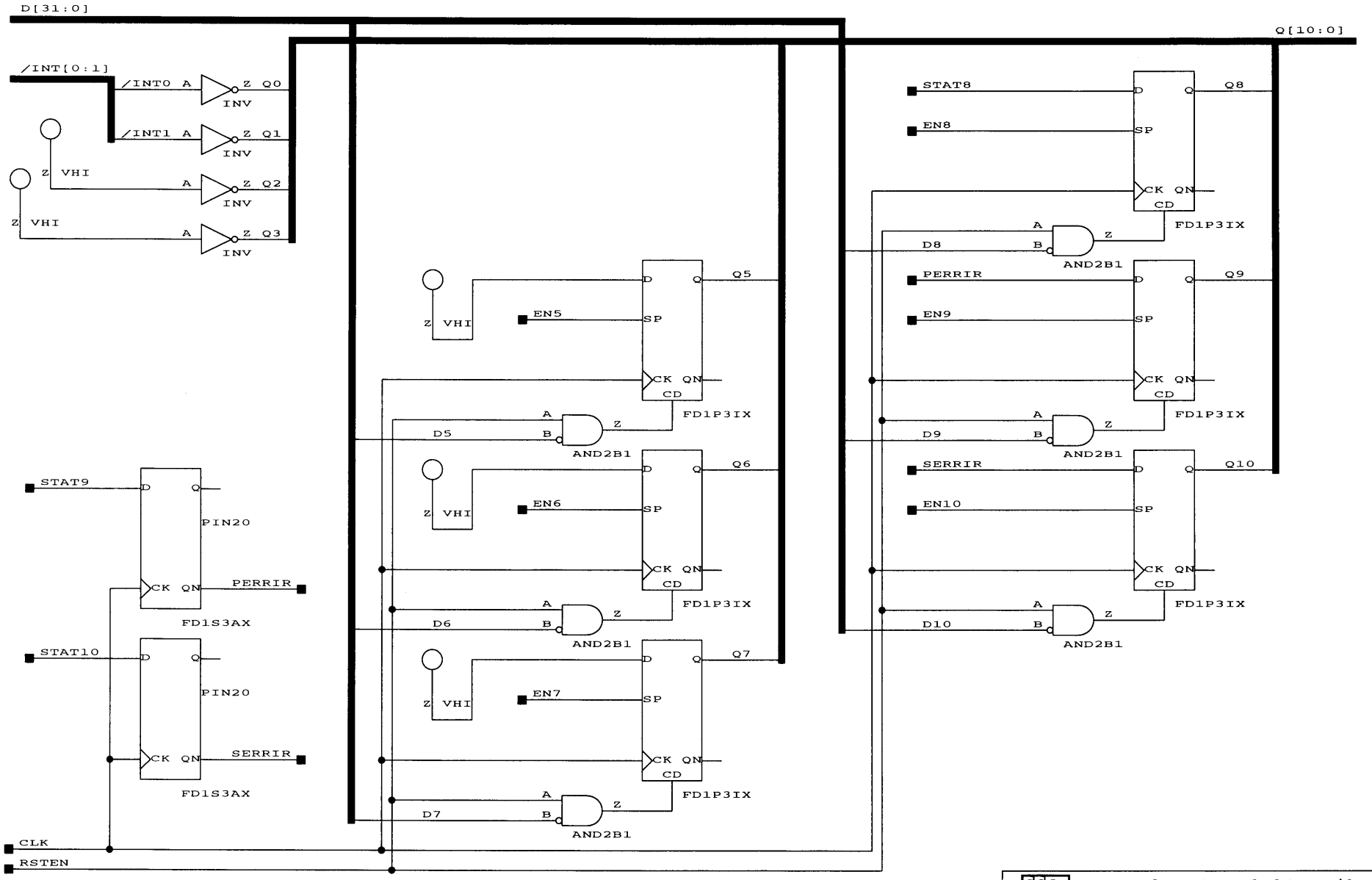

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI STATUS REGISTER |
| Issue 2 | | |
| Issue 3 | | File: smistat Page:1 of 1 |



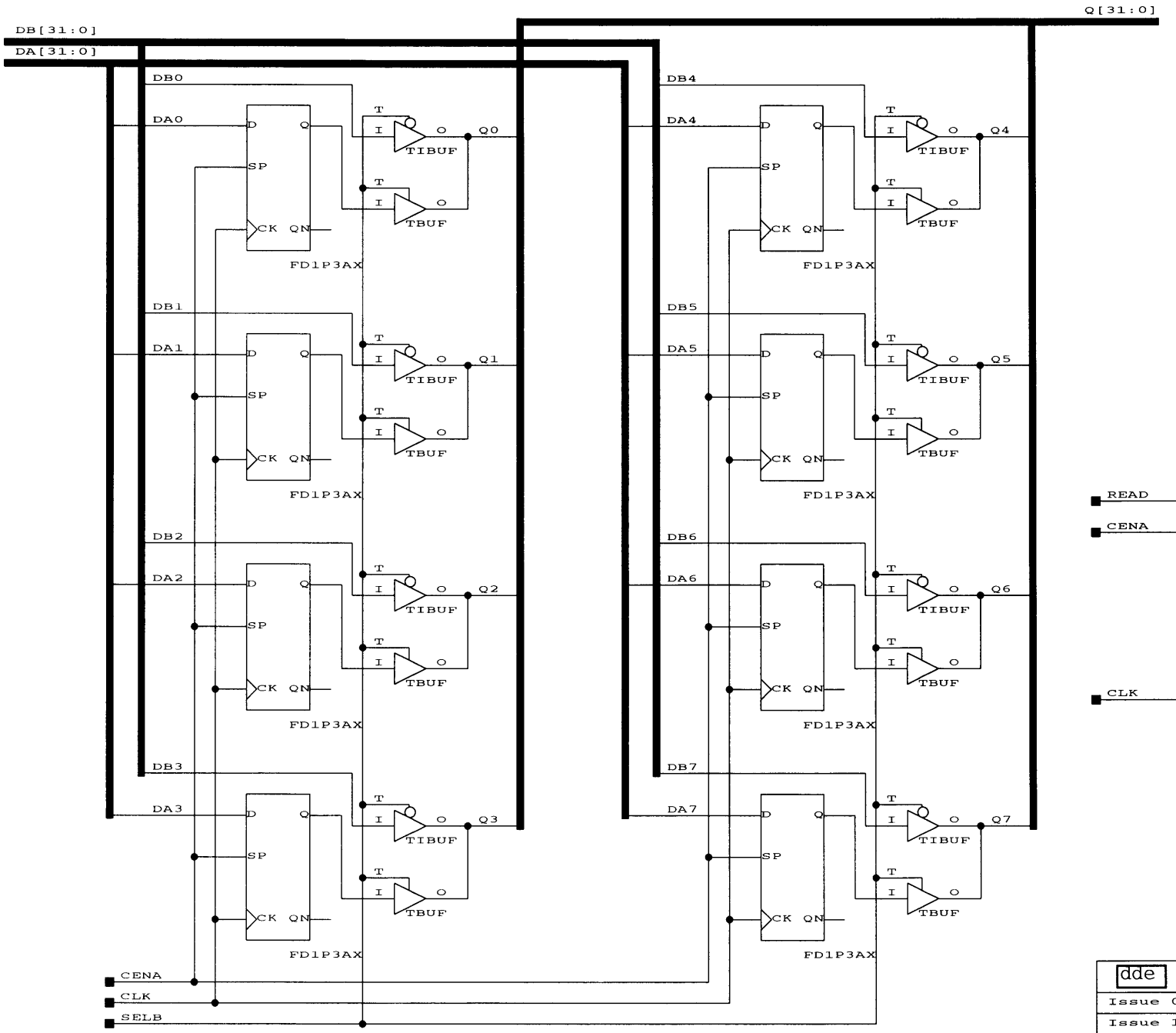

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>SMI CONTROL REGISTER |
| Issue 1 | | |
| Issue 2 | | |
| Issue 3 | | File: smicontr Page:1 of 1 |

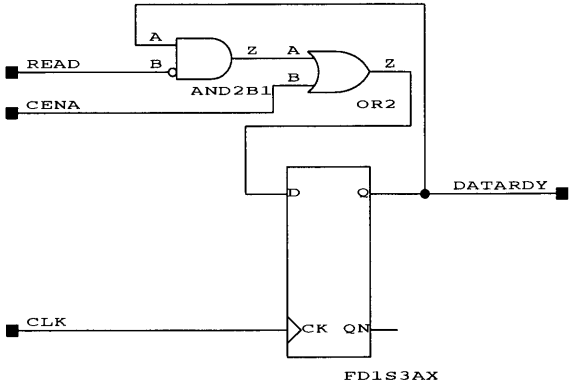



| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>PCI CONTROL REGISTER | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: pcicontr | Page:1 of 1 |

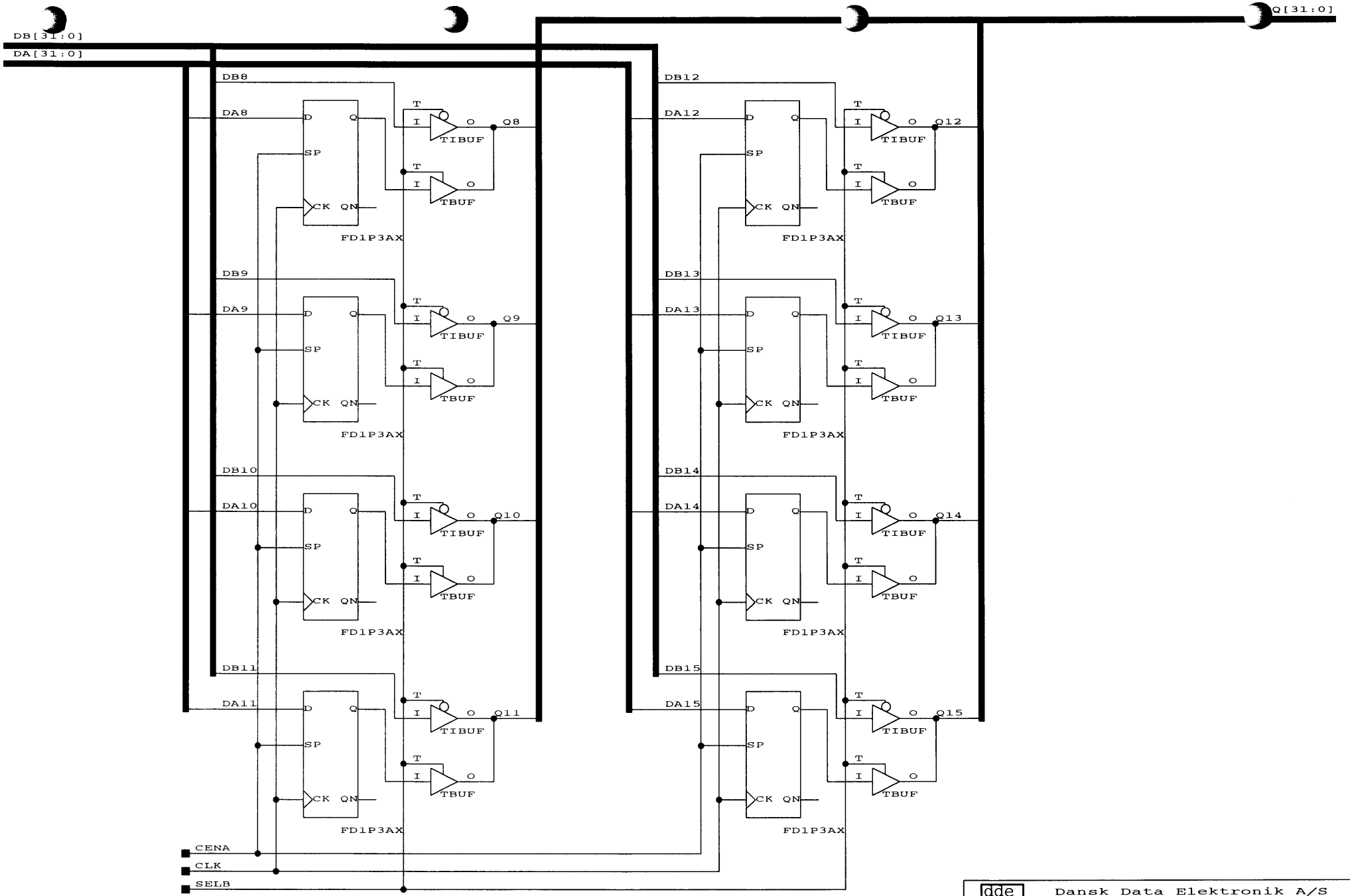



| | | |
|---|---|---|
| dde | Dansk Data Elektronik A/S | |
| Issue 0 | 970917 | BSP301-1 |
| Issue 1 | | PCI STATUS REGISTER |
| Issue 2 | | |
| Issue 3 | | File: pcistat Page:1 of 1 |

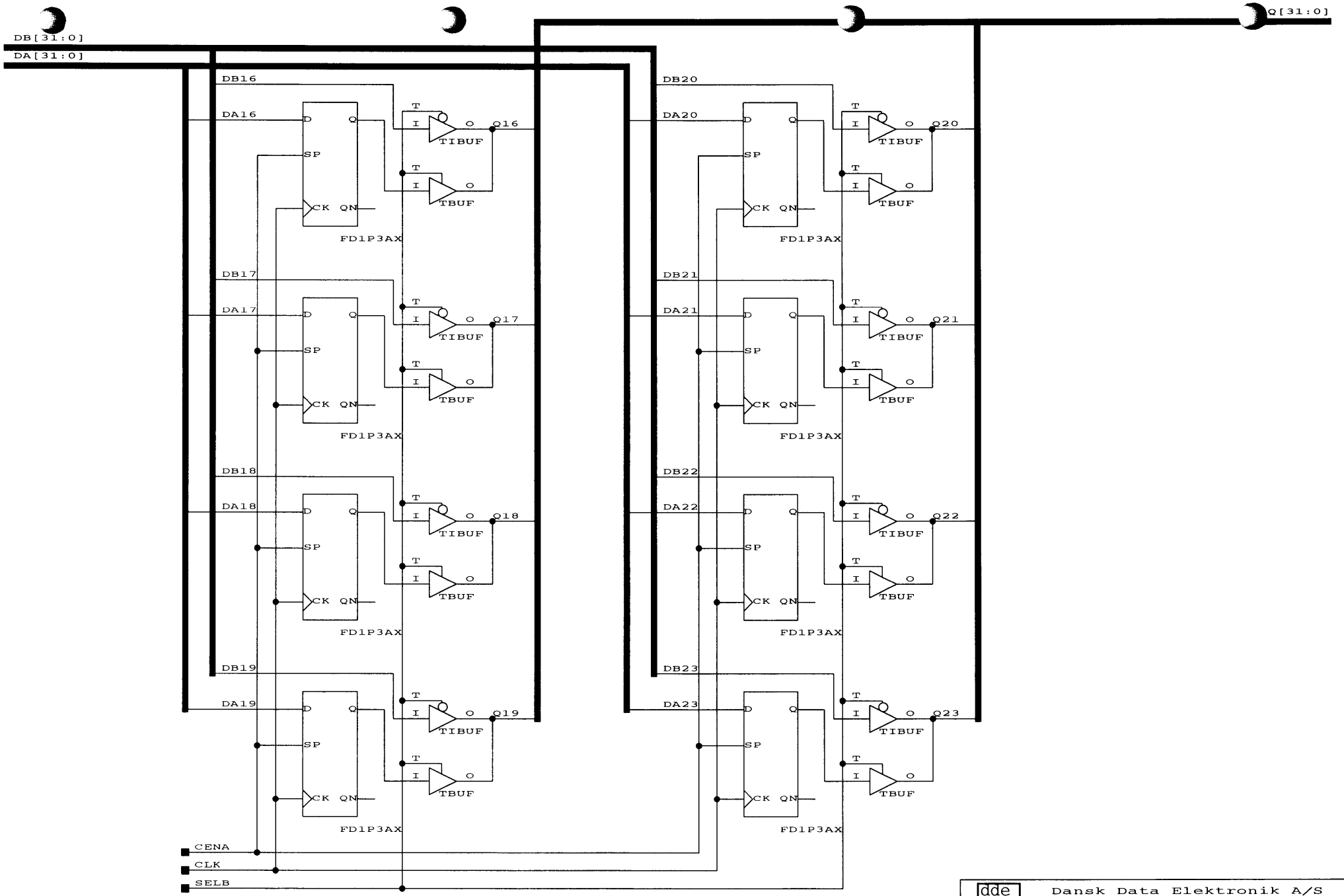


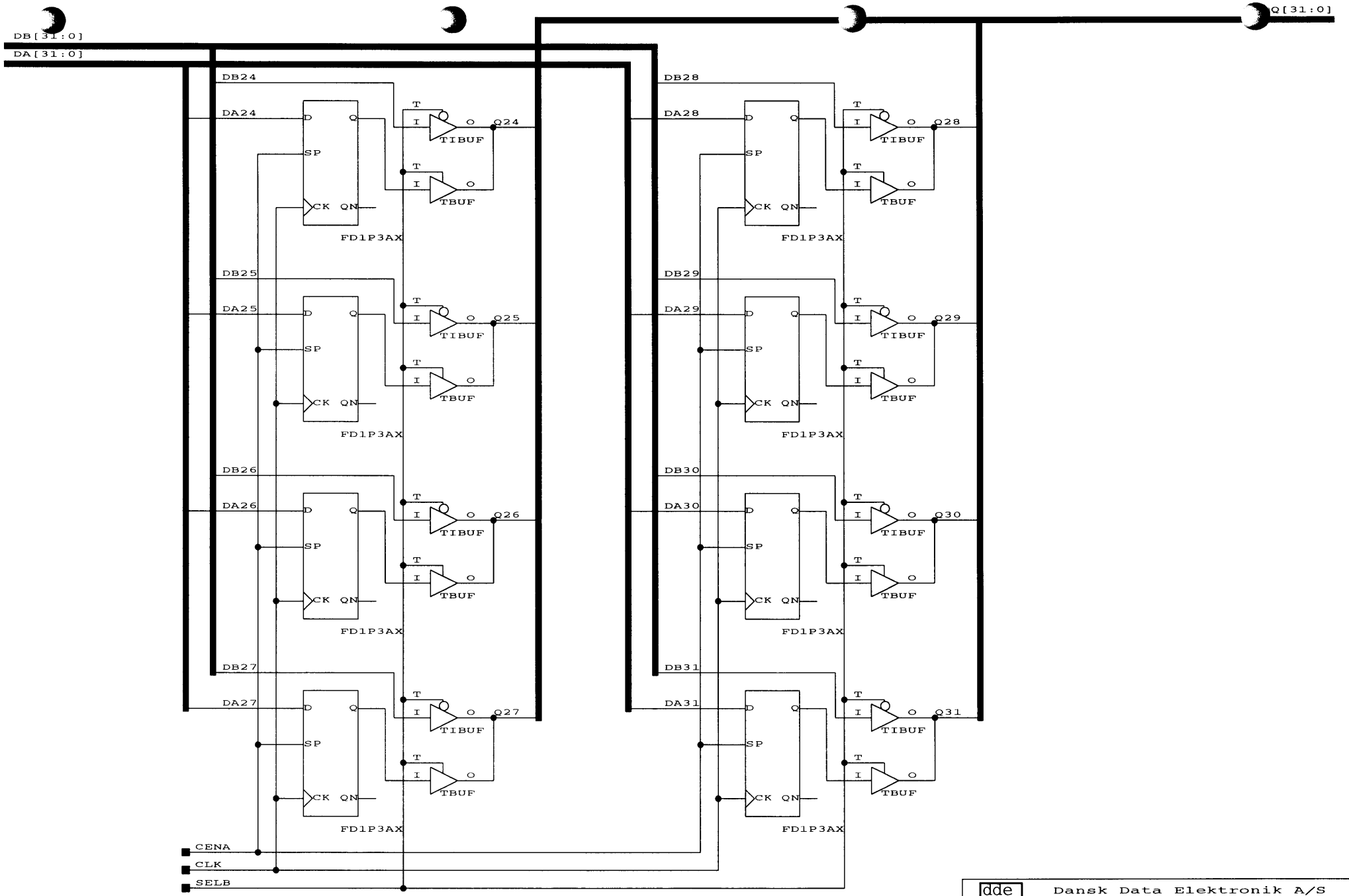

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI DATA REGISTER WITH MUX |
| Issue 2 | | |
| Issue 3 | | File: pcidreg Page:1 of 4 |



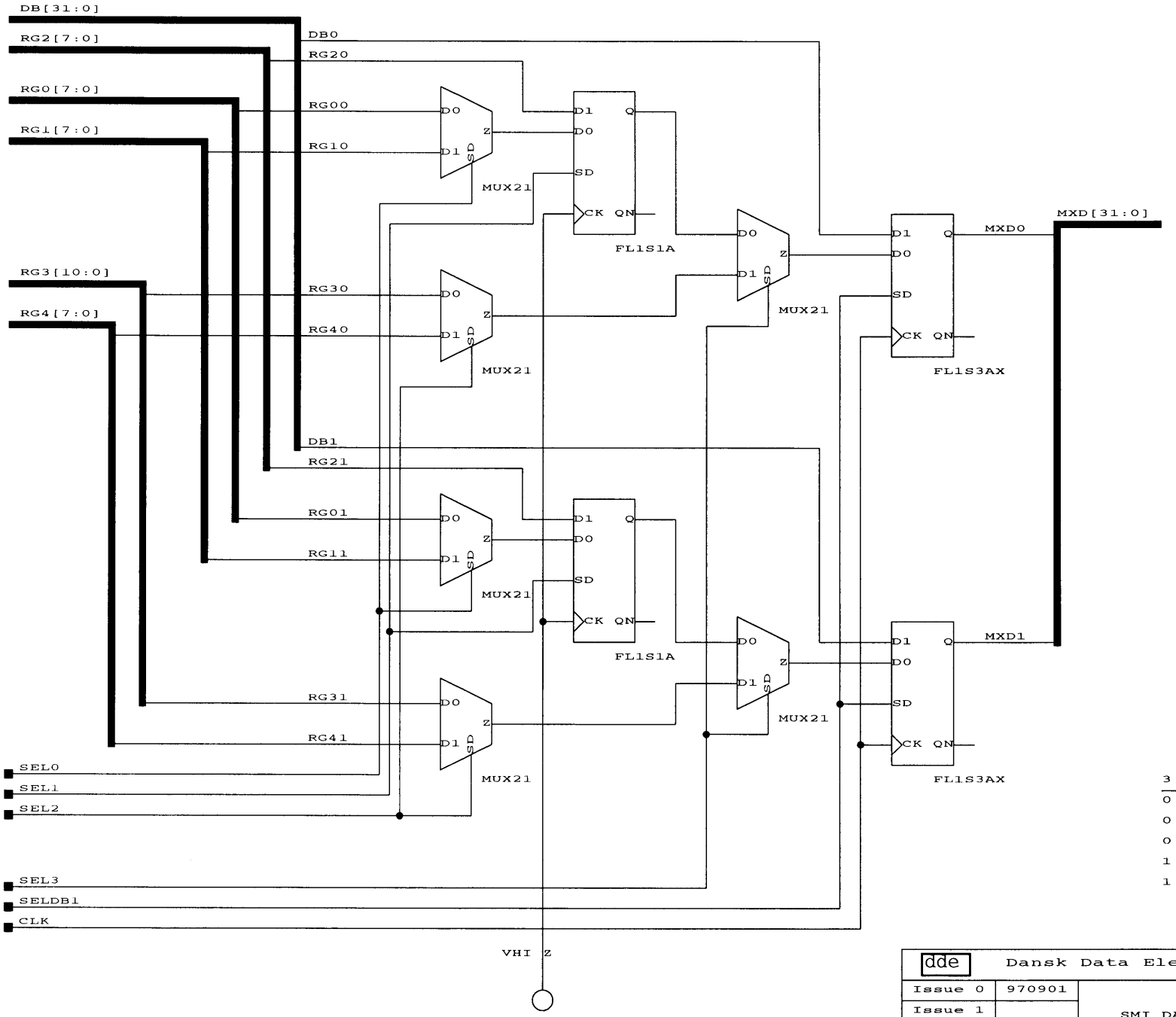

| | | | |
|---|---|---|---|
| dde | Dansk Data Elektronik A/S | | |
| Issue 0 | 970901 | BSP301-1 | |
| Issue 1 | | PCI DATA REGISTER WITH MUX | |
| Issue 2 | | | |
| Issue 3 | | File: pcidreg | Page:2 of 4 |




| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI DATA REGISTER WITH MUX |
| Issue 2 | | |
| Issue 3 | | File: pcidreg Page:3 of 4 |

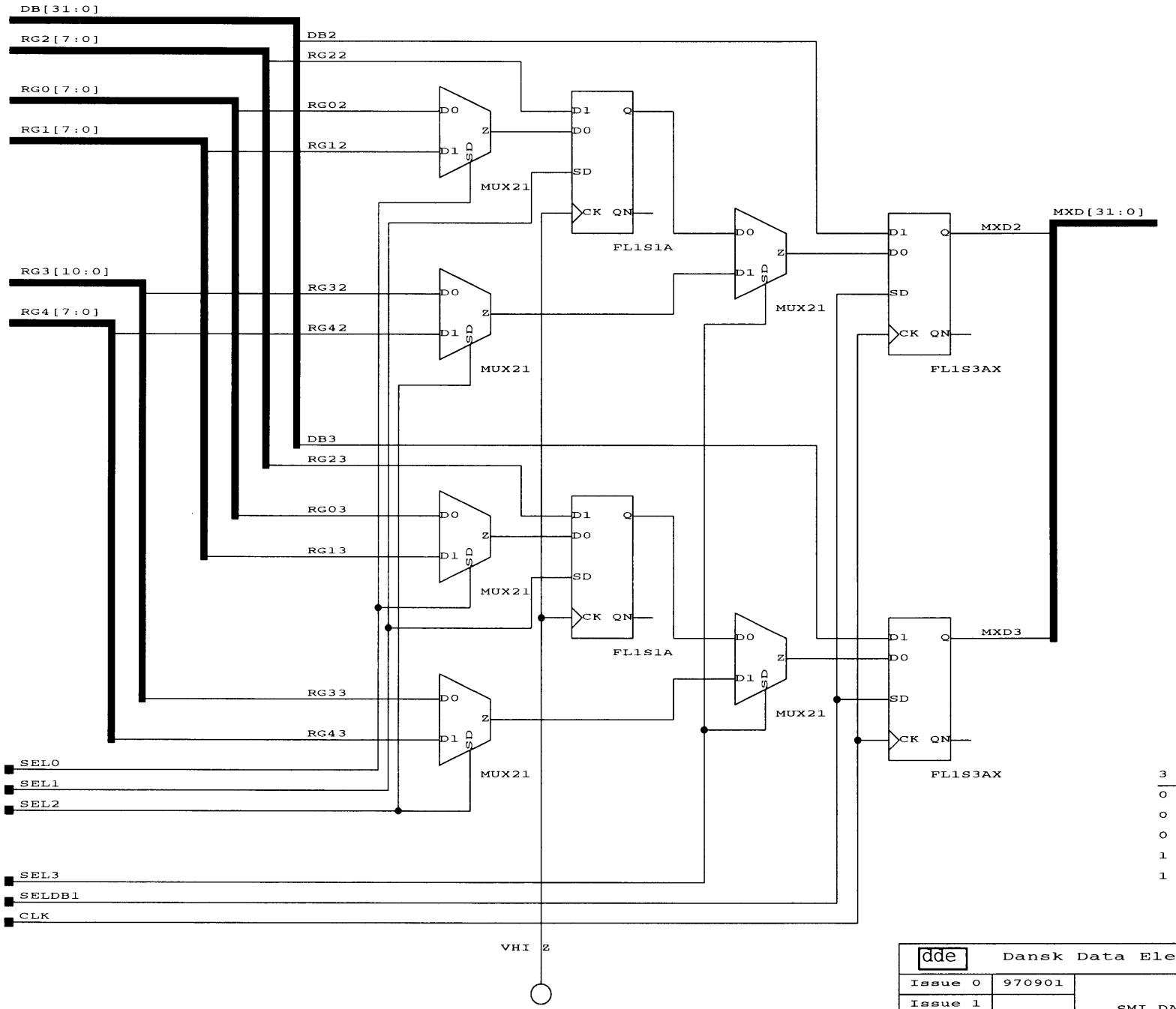



| | | |
|---|---|---|
| dde | Dansk Data Elektronik A/S | |
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI DATA REGISTER WITH MUX |
| Issue 2 | | |
| Issue 3 | | File: pcidreg Page:4 of 4 |



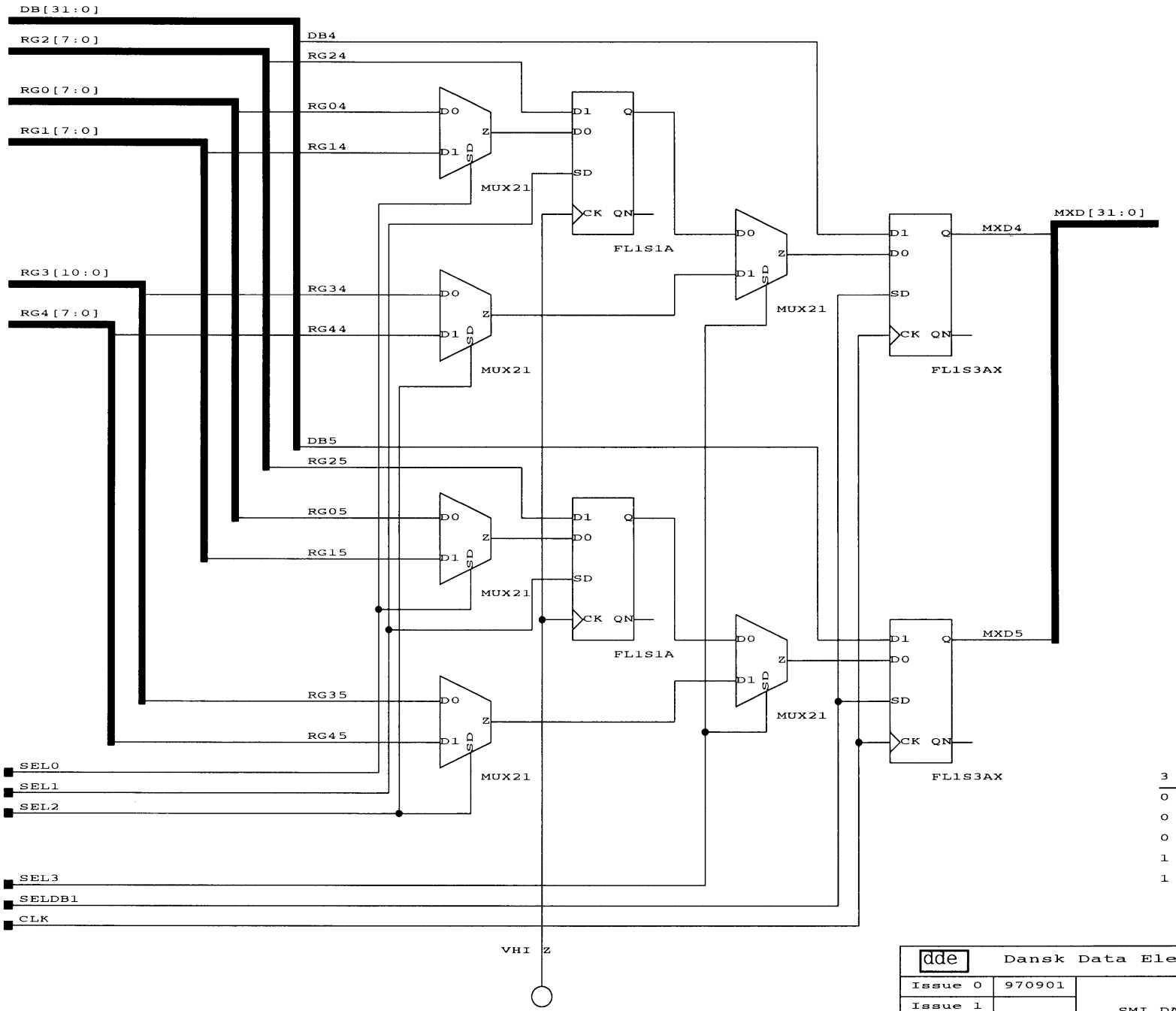

| SEL 3 | 2 | 1 | 0 | SELECTED |
|---|---|---|---|---|
| 0 | / | 0 | 0 | RG0 |
| 0 | / | 0 | 1 | RG1 |
| 0 | / | 1 | / | RG2 |
| 1 | 0 | / | / | RG3 |
| 1 | 1 | / | / | RG4 |

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 SMI DATA MULTIPLEXER | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: smidmux | Page:1 of 8 |



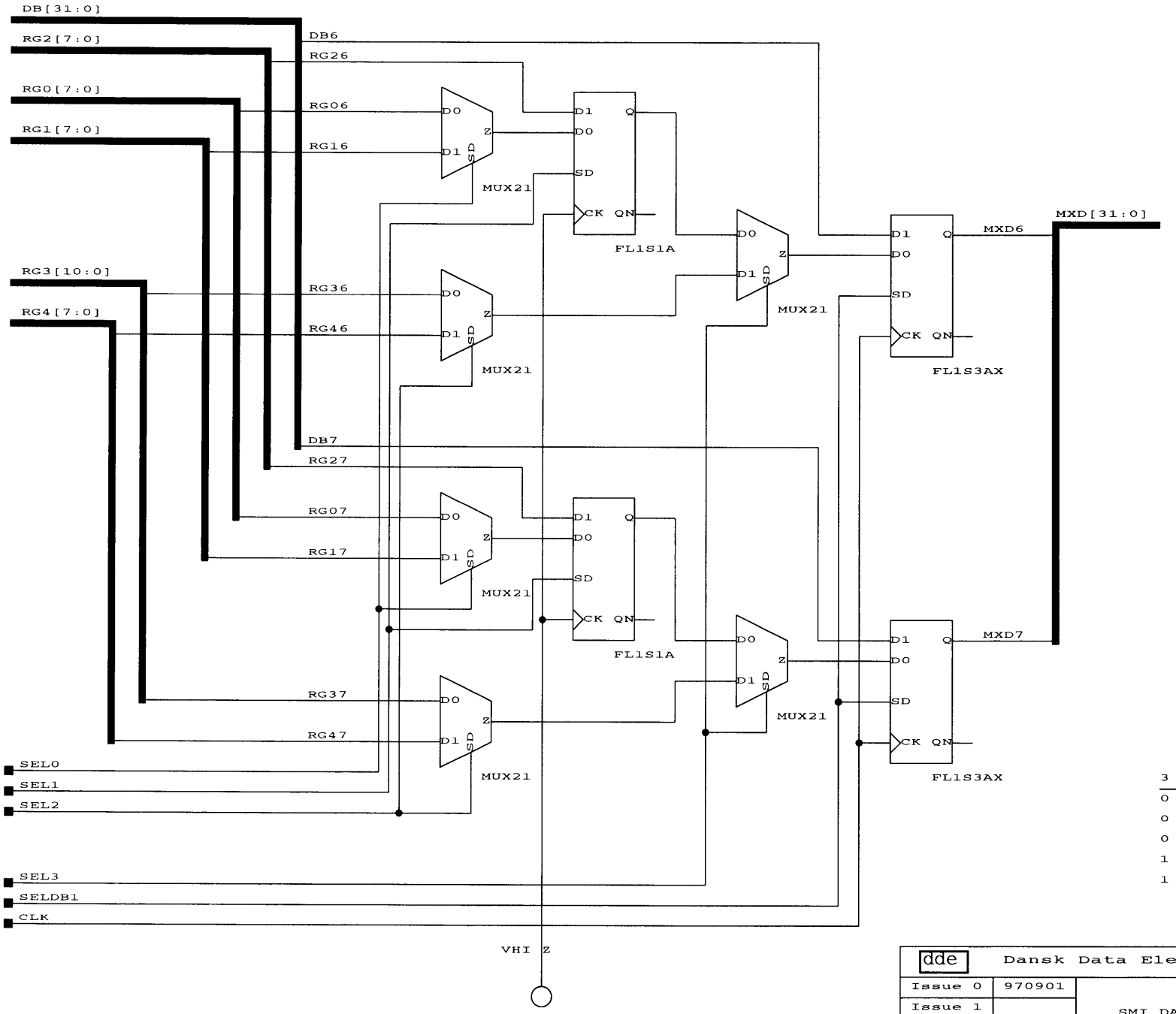

| SEL 3 | 2 | 1 | 0 | SELECTED |
|---|---|---|---|---|
| 0 | / | 0 | 0 | RG0 |
| 0 | / | 0 | 1 | RG1 |
| 0 | / | 1 | / | RG2 |
| 1 | 0 | / | / | RG3 |
| 1 | 1 | / | / | RG4 |

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 SMI DATA MULTIPLEXER | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: smidmux | Page: 2 of 8 |



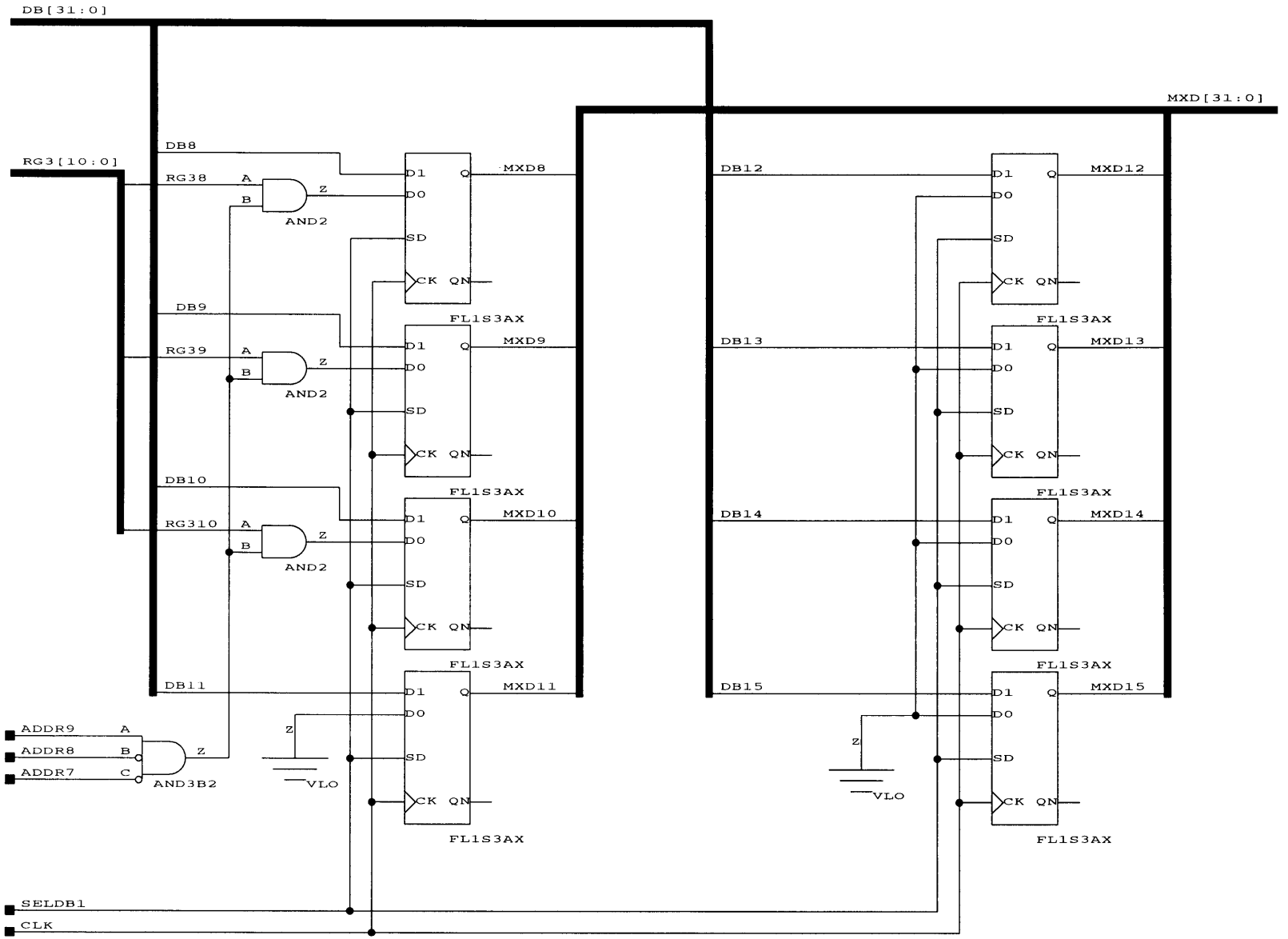

| SEL 3 | 2 | 1 | 0 | SELECTED |
|---|---|---|---|---|
| 0 | / | 0 | 0 | RG0 |
| 0 | / | 0 | 1 | RG1 |
| 0 | / | 1 | / | RG2 |
| 1 | 0 | / | / | RG3 |
| 1 | 1 | / | / | RG4 |

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 SMI DATA MULTIPLEXER |
| Issue 1 | | |
| Issue 2 | | |
| Issue 3 | | File: smidmux Page: 3 of 8 |

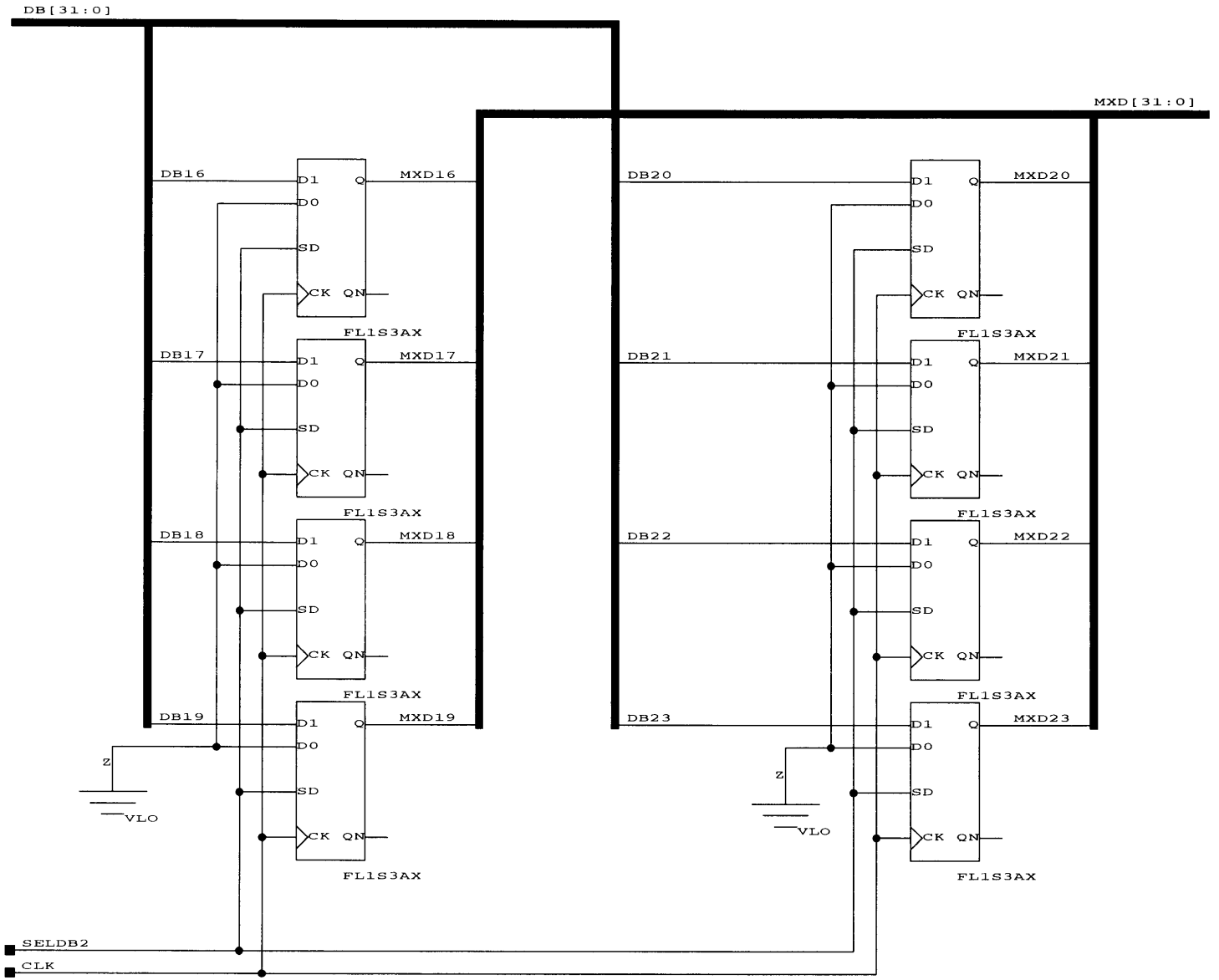



| SEL 3 | 2 | 1 | 0 | SELECTED |
|---|---|---|---|---|
| 0 | / | 0 | 0 | RG0 |
| 0 | / | 0 | 1 | RG1 |
| 0 | / | 1 | / | RG2 |
| 1 | 0 | / | / | RG3 |
| 1 | 1 | / | / | RG4 |

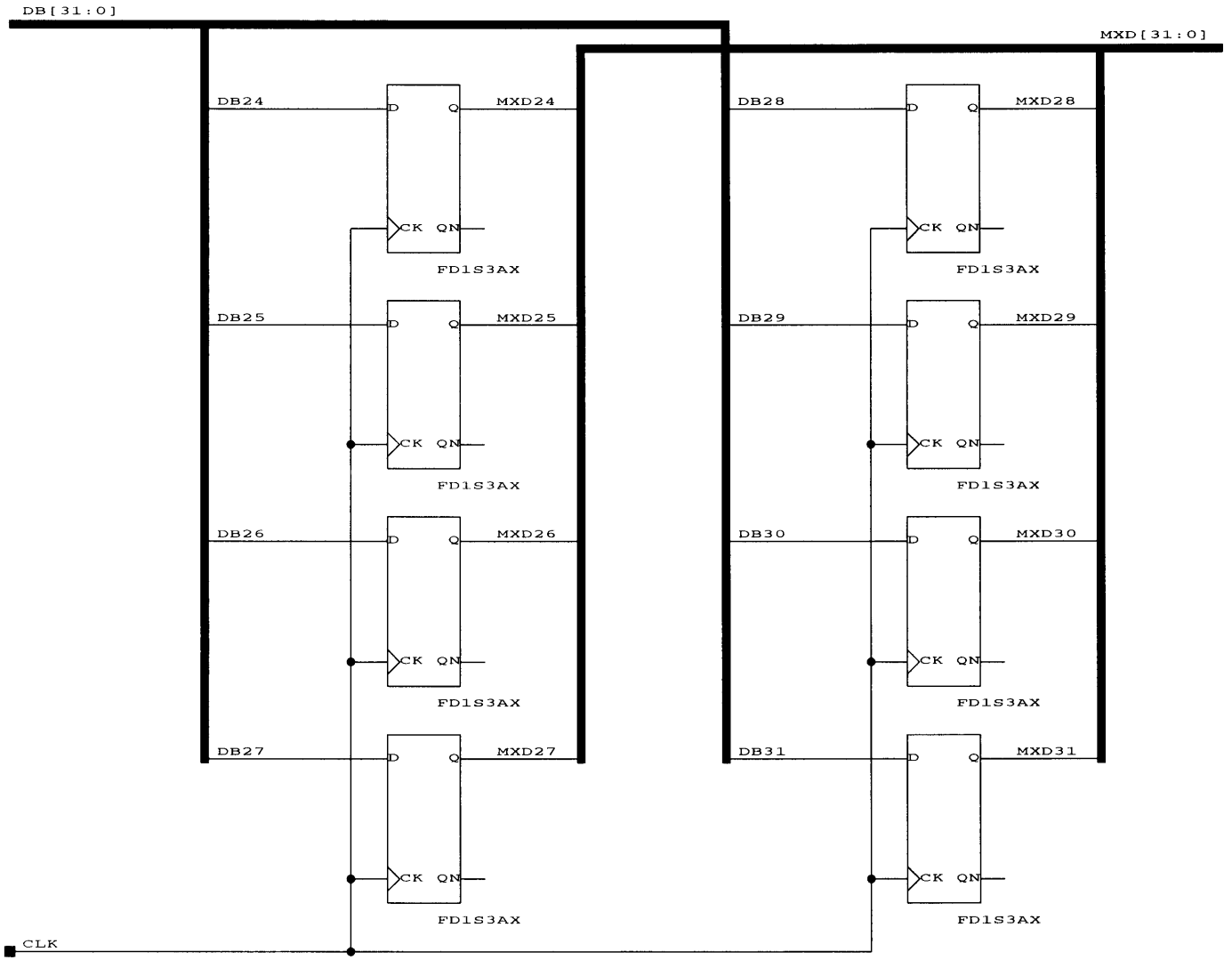
| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 SMI DATA MULTIPLEXER | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: smidmux | Page:4 of 8 |



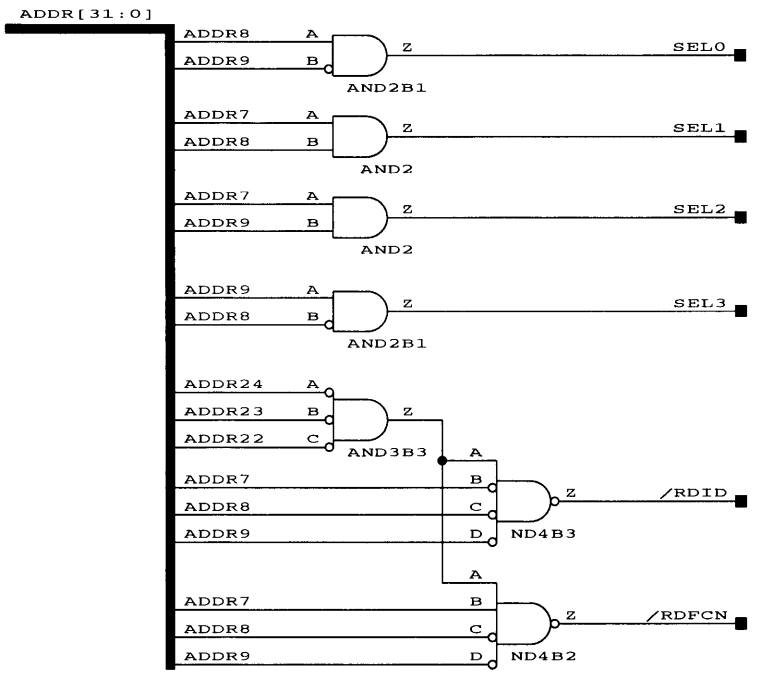

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI DATA MULTIPLEXER |
| Issue 2 | | |
| Issue 3 | | File: smidmux Page:5 of 8 |




| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI DATA MULTIPLEXER |
| Issue 2 | | |
| Issue 3 | | File: smidmux Page:6 of 8 |

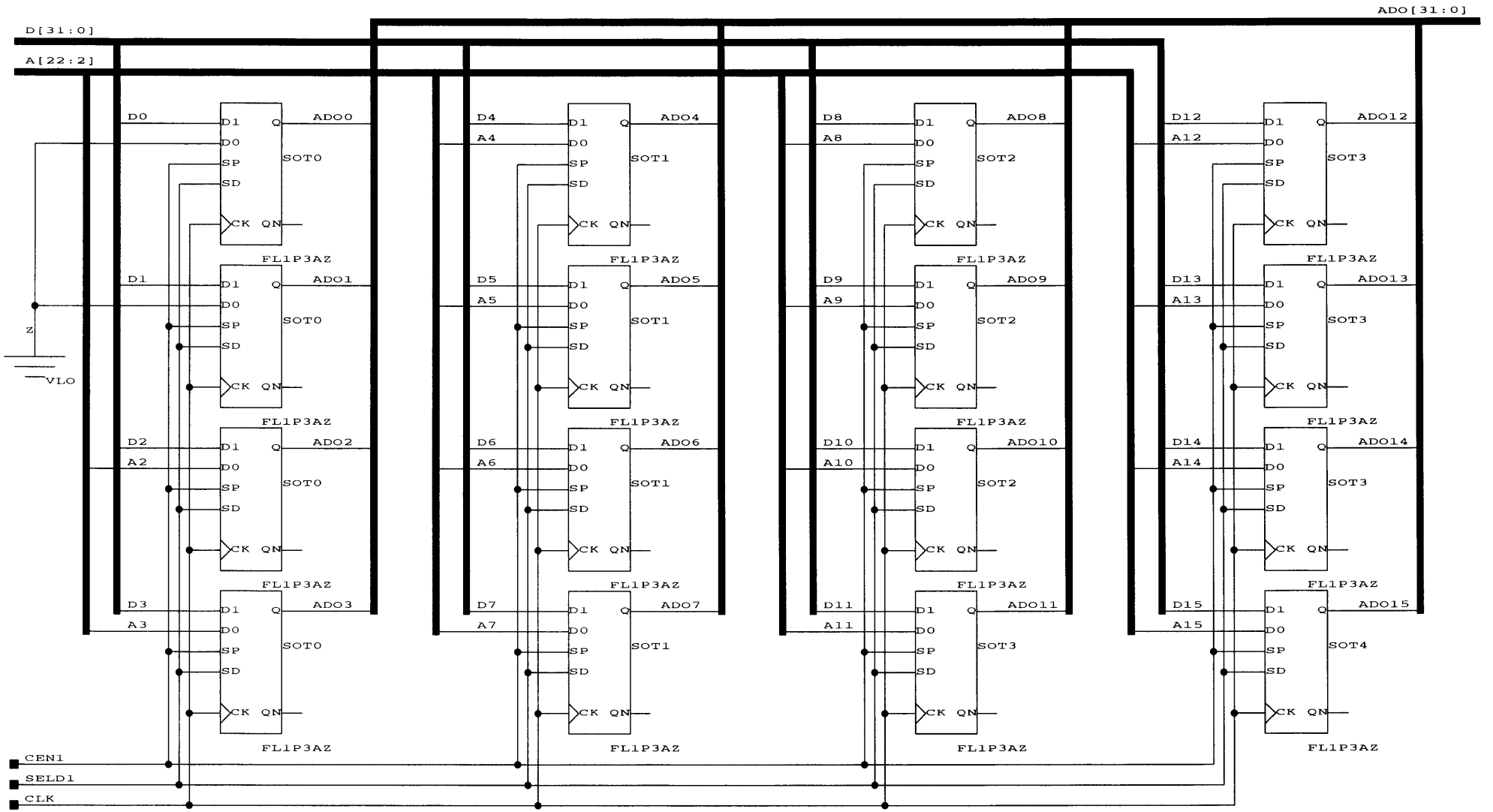



| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>SMI DATA MULTIPLEXER | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: smidmux | Page:7 of 8 |

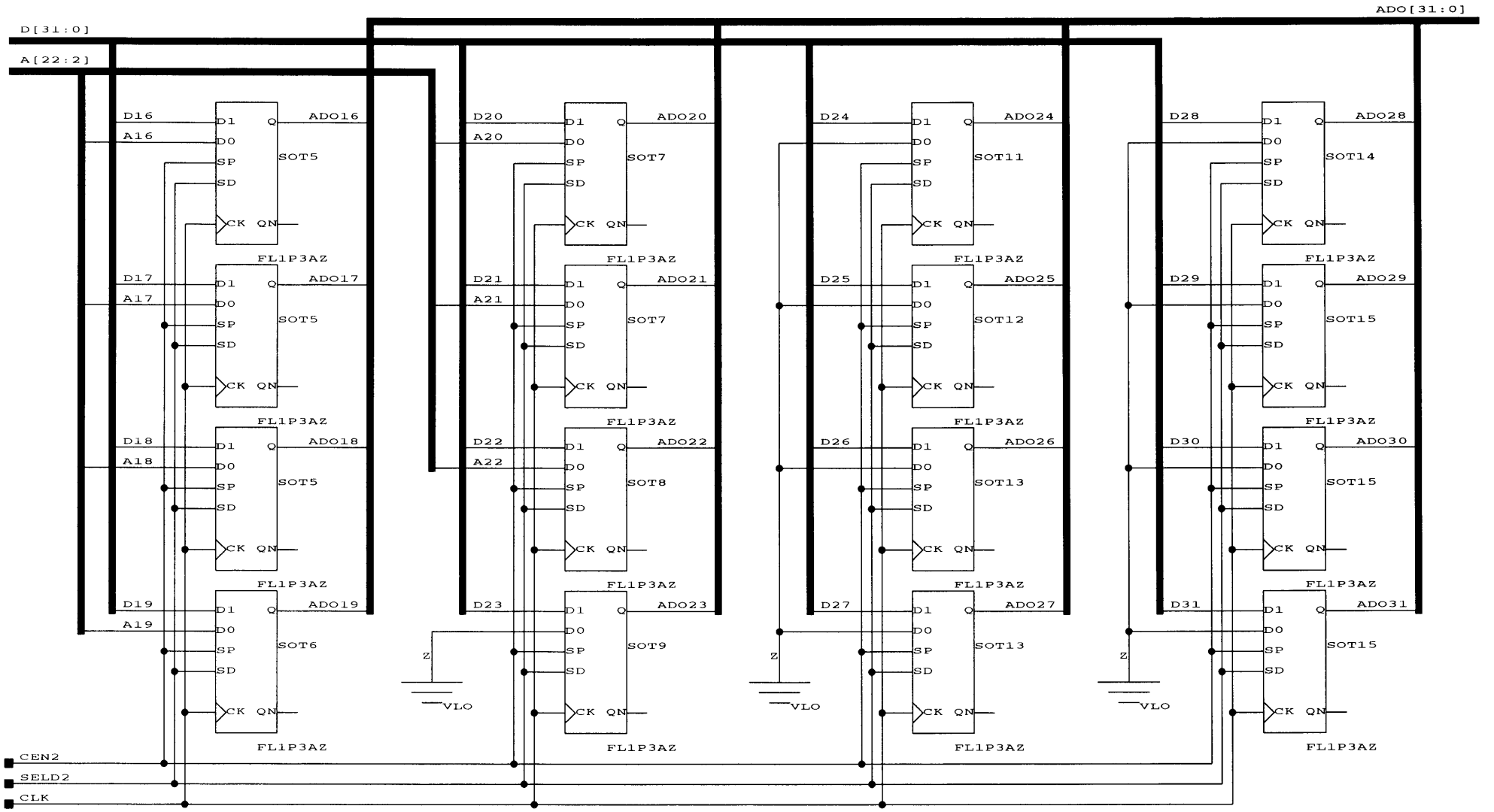



| SEL 3 | 2 | 1 | 0 | SELECTED |
|---|---|---|---|---|
| 0 | / | 0 | 0 | RG0 |
| 0 | / | 0 | 1 | RG1 |
| 0 | / | 1 | / | RG2 |
| 1 | 0 | / | / | RG3 |
| 1 | 1 | / | / | RG4 |

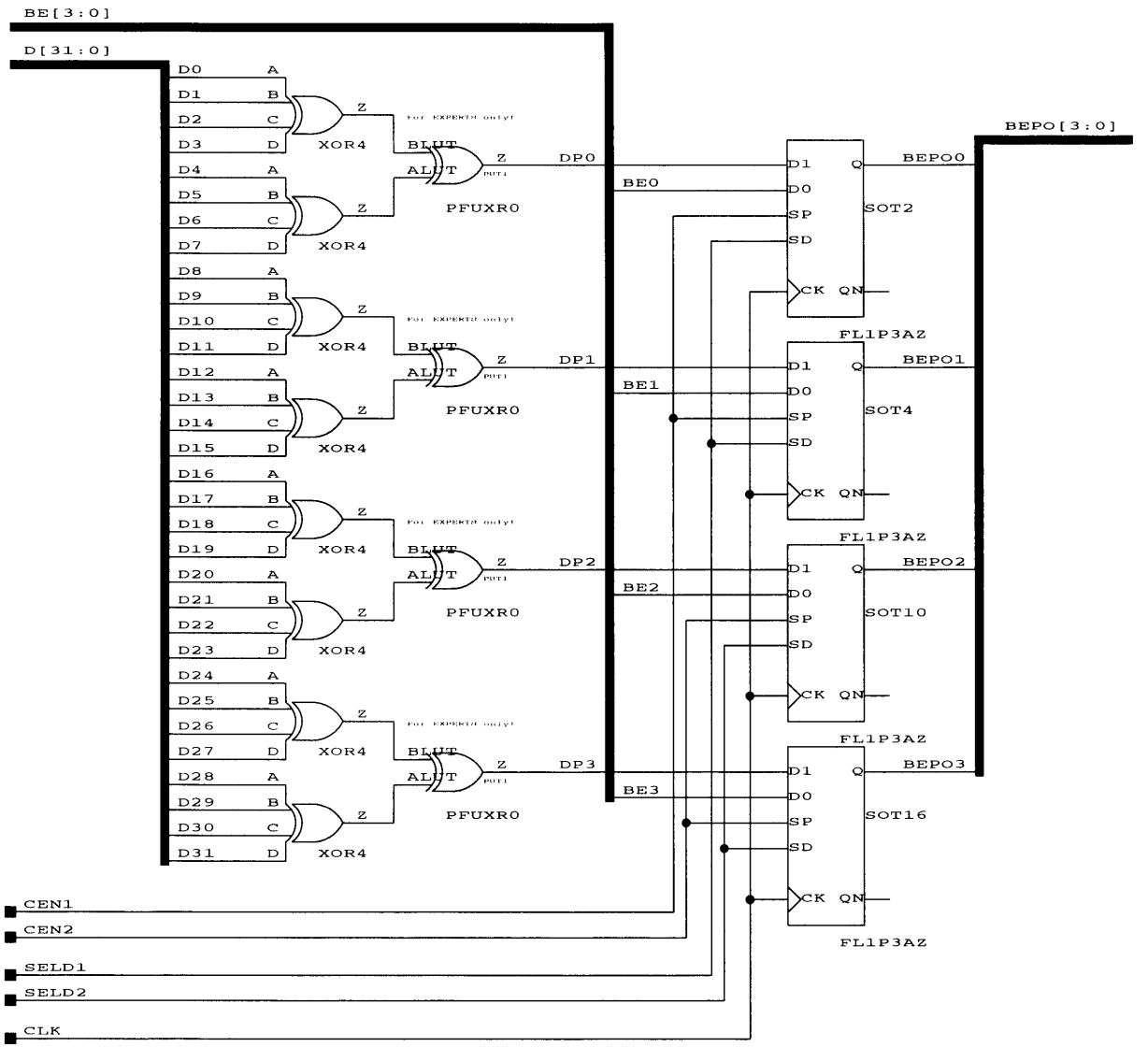
| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 SMI DATA MULTIPLEXER | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: smidmux | Page:8 of 8 |



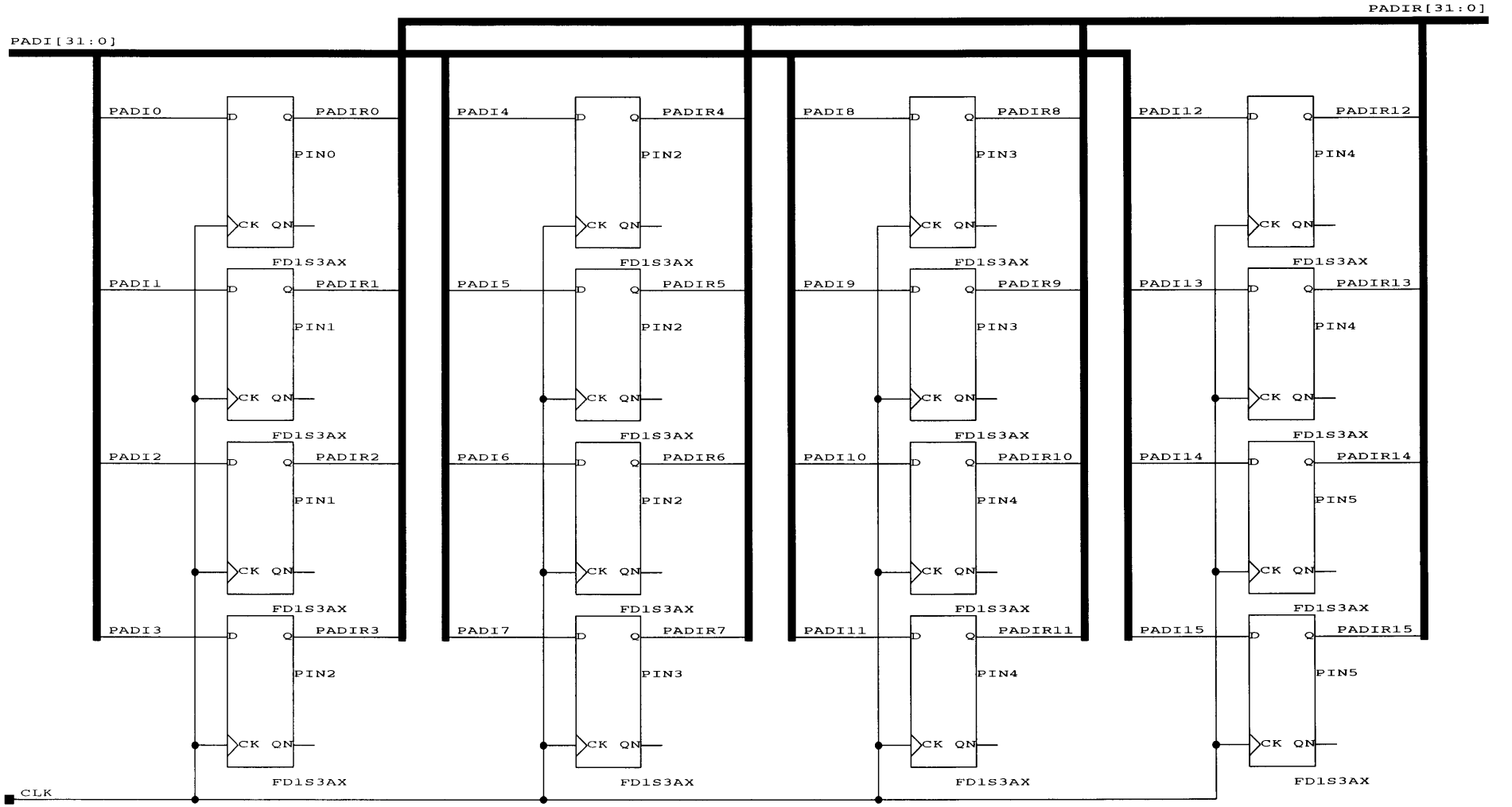

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI ADDR/DATA OUTPUT MULTIPLEXER & REGISTER |
| Issue 2 | | |
| Issue 3 | | File: smiadmux Page:1 of 3 |



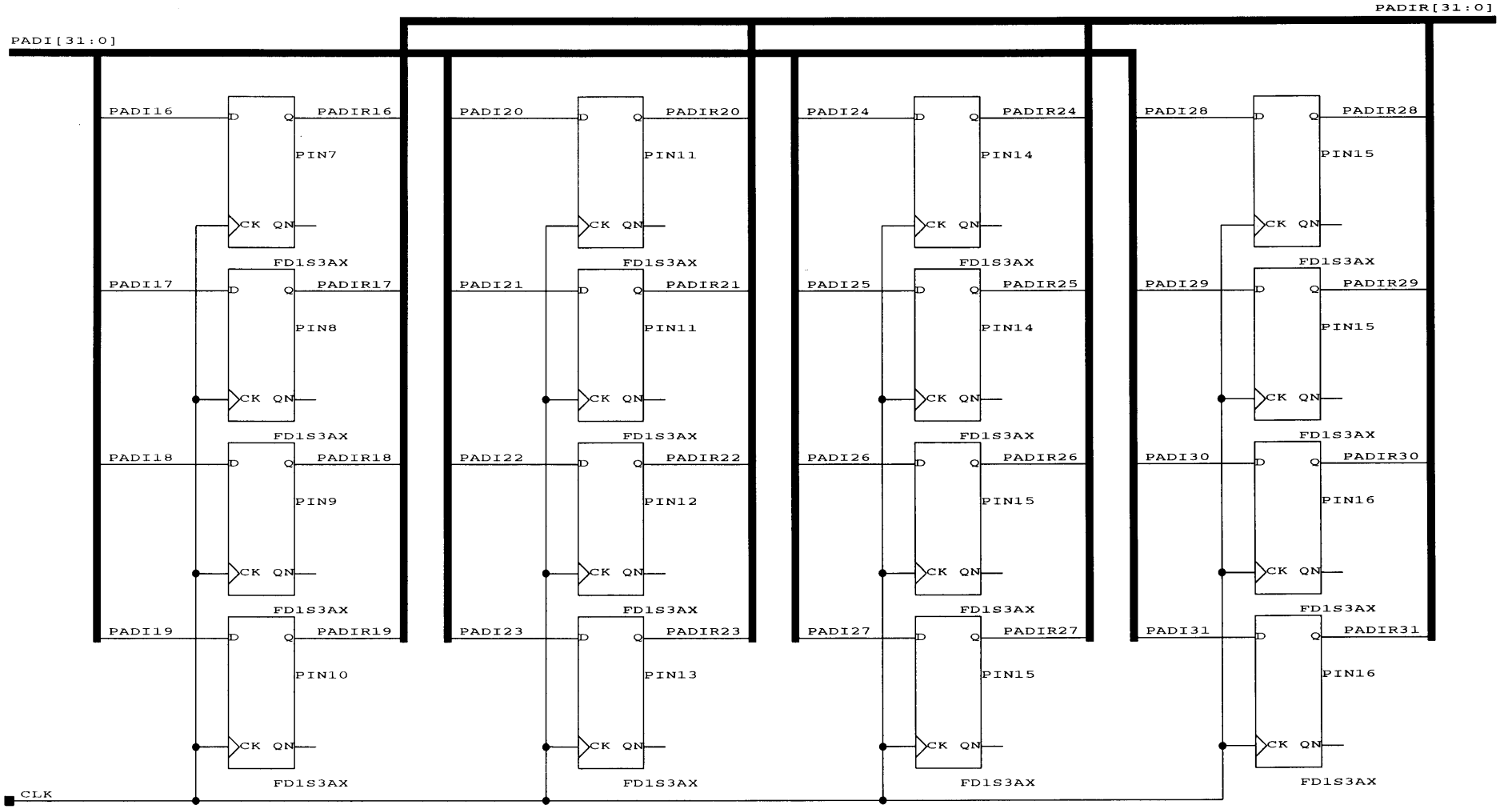

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI ADDR/DATA OUTPUT |
| Issue 2 | | MULTIPLEXER & REGISTER |
| Issue 3 | | File: smiadmux Page:2 of 3 |



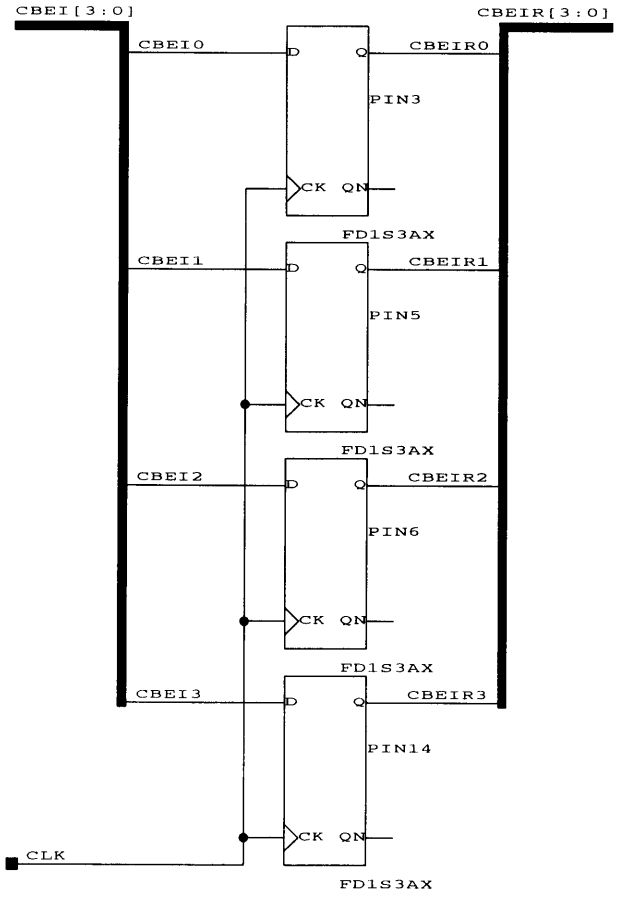

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>SMI ADDR/DATA OUTPUT<br>MULTIPLEXER & REGISTER | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: smiadmux | Page:3 of 3 |



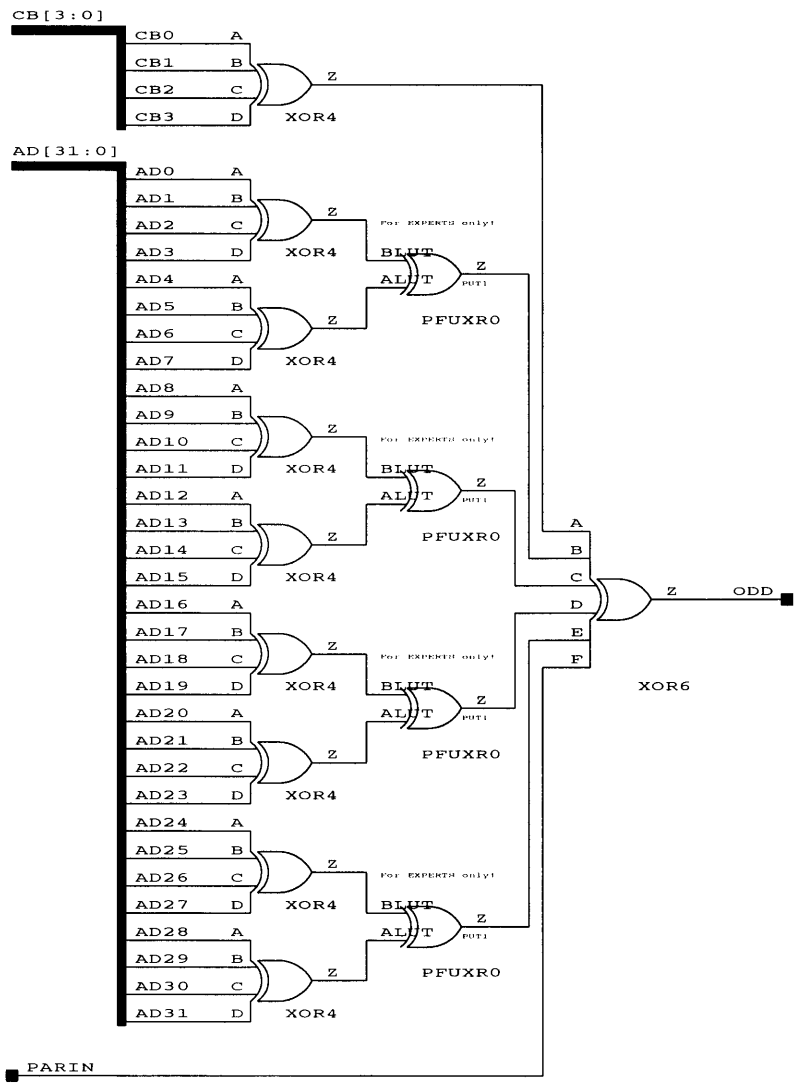

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 | |
| Issue 1 | | PCI INPUT REGISTER | |
| Issue 2 | | | |
| Issue 3 | | File: pinreg | Page:1 of 3 |



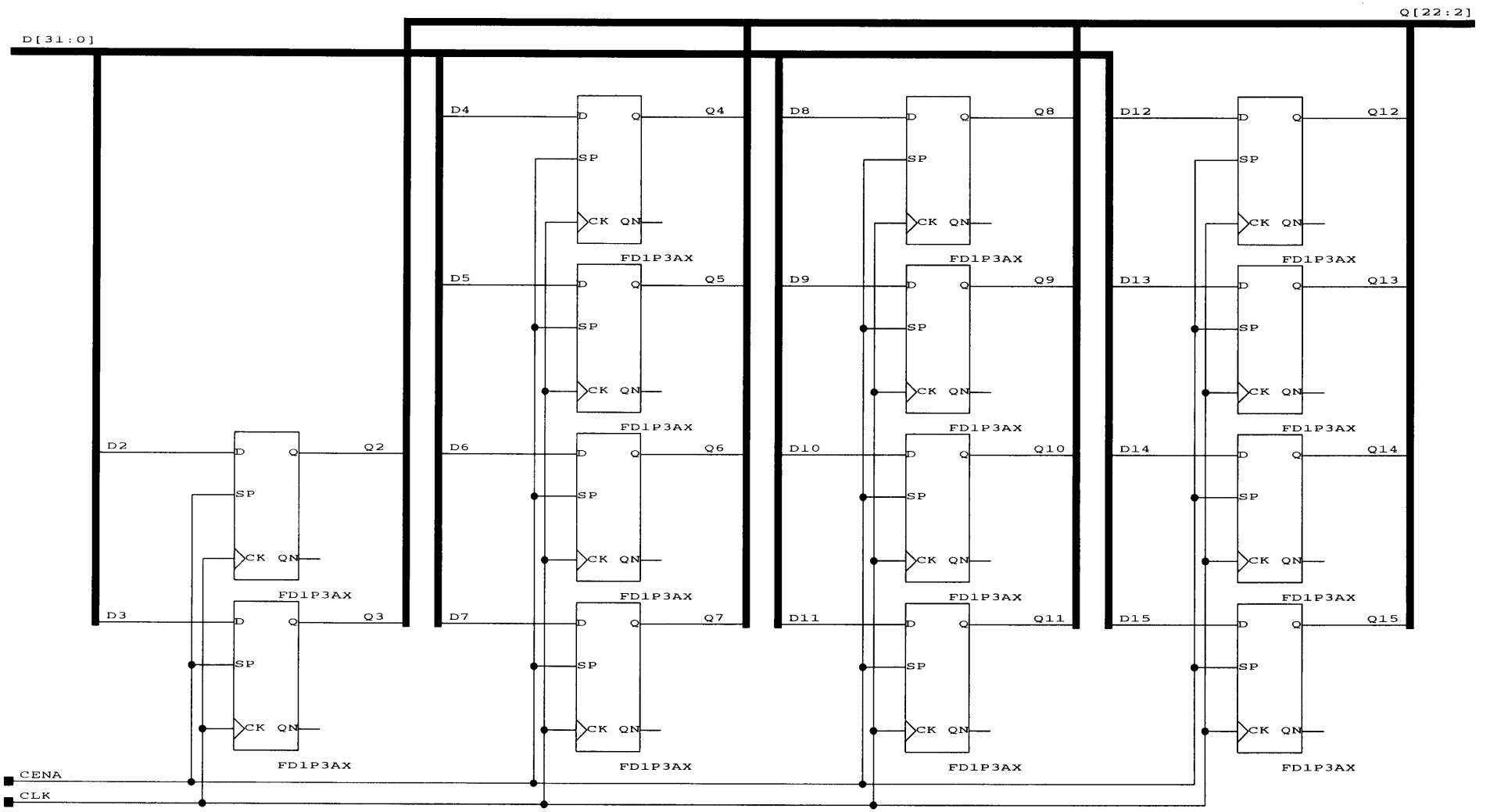

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI INPUT REGISTER |
| Issue 2 | | |
| Issue 3 | | File: pinreg Page:2 of 3 |



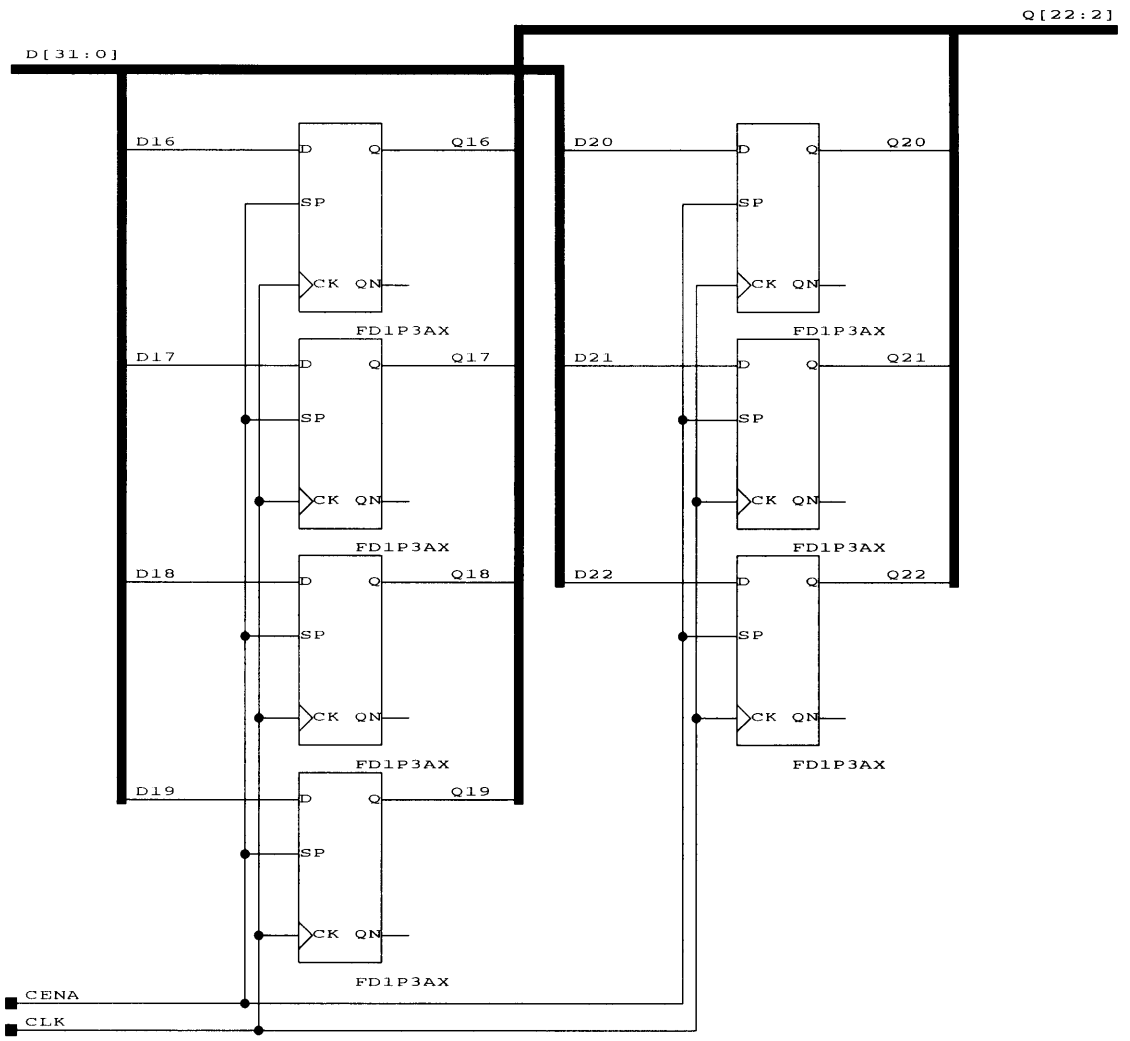

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>PCI INPUT REGISTER | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: pinreg | Page:3 of 3 |



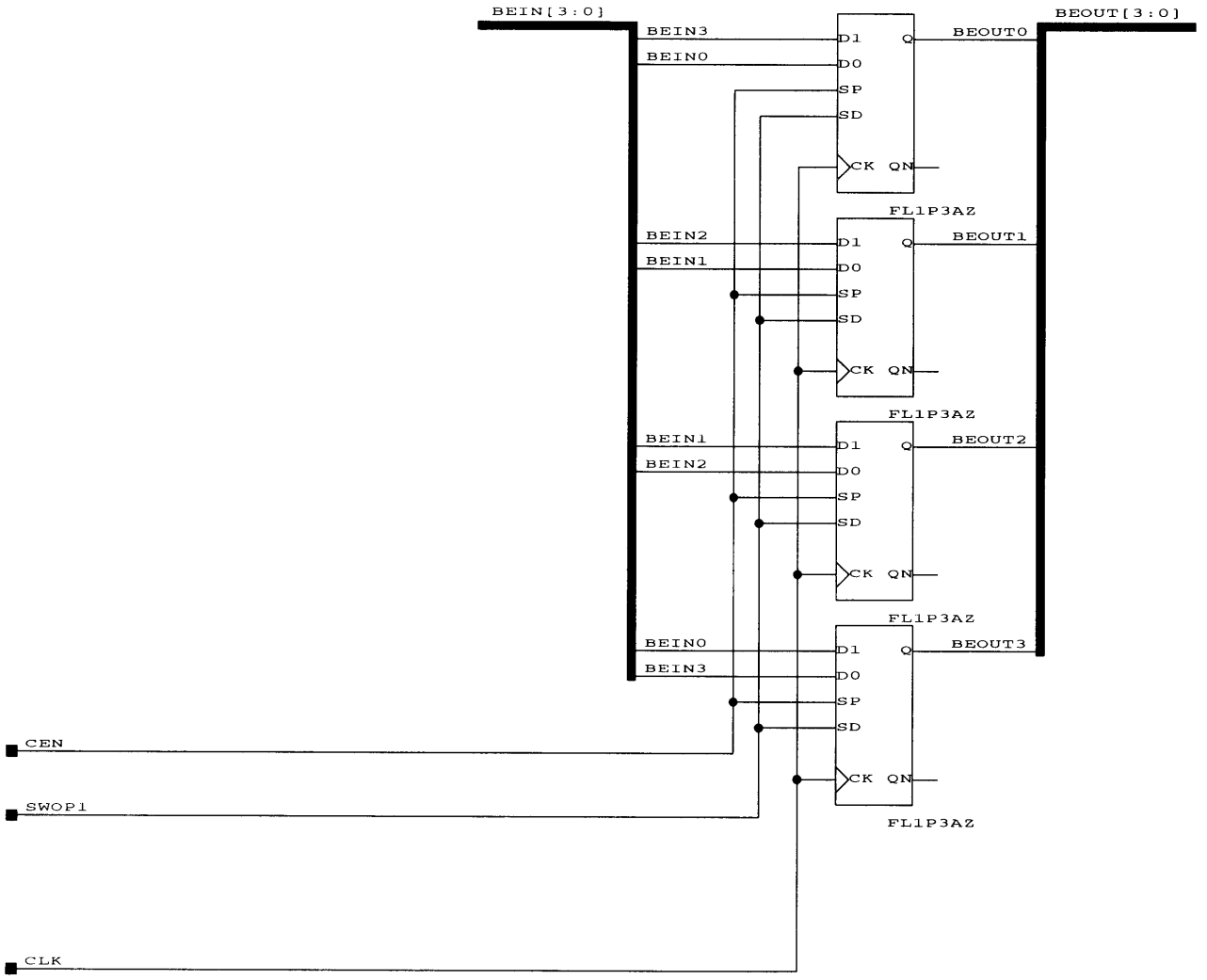

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>PCI PARITY CHECK | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: pcipchk | Page:1 of 1 |



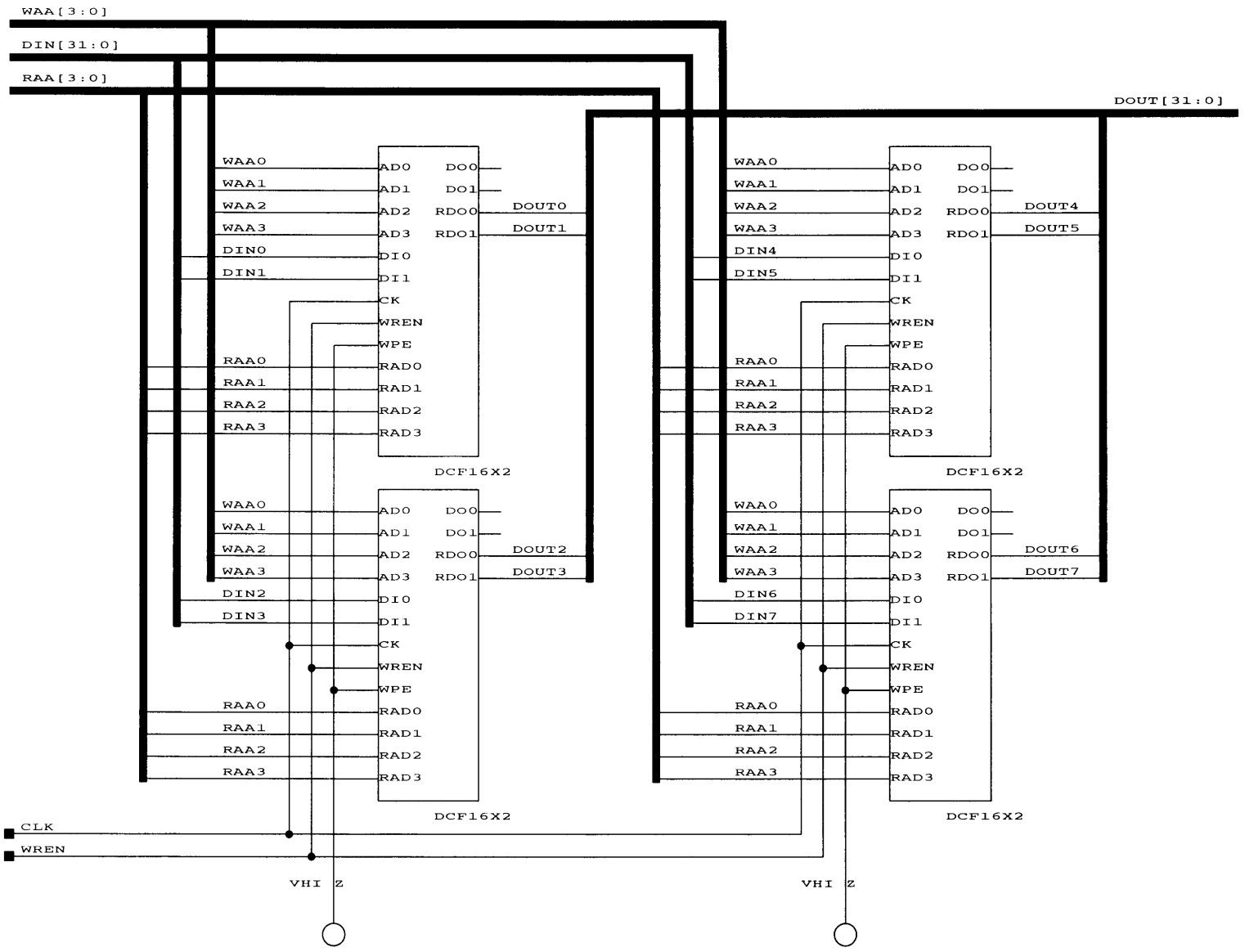

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI ADDRESS INPUT REGISTER |
| Issue 2 | | |
| Issue 3 | | File: pciaddrin Page:1 of 2 |



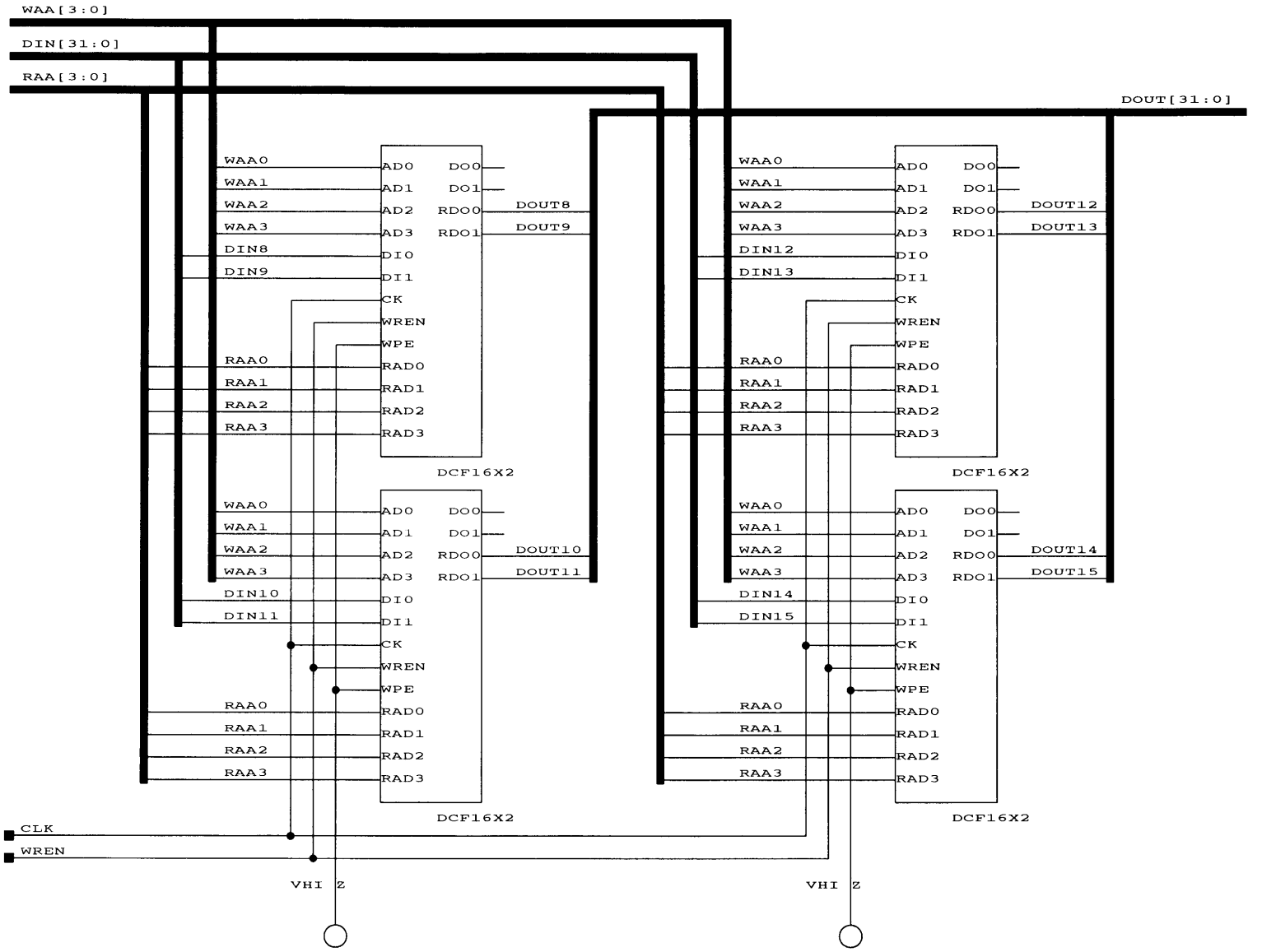

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI ADDRESS INPUT REGISTER |
| Issue 2 | | |
| Issue 3 | | File: pciaddrin Page:2 of 2 |



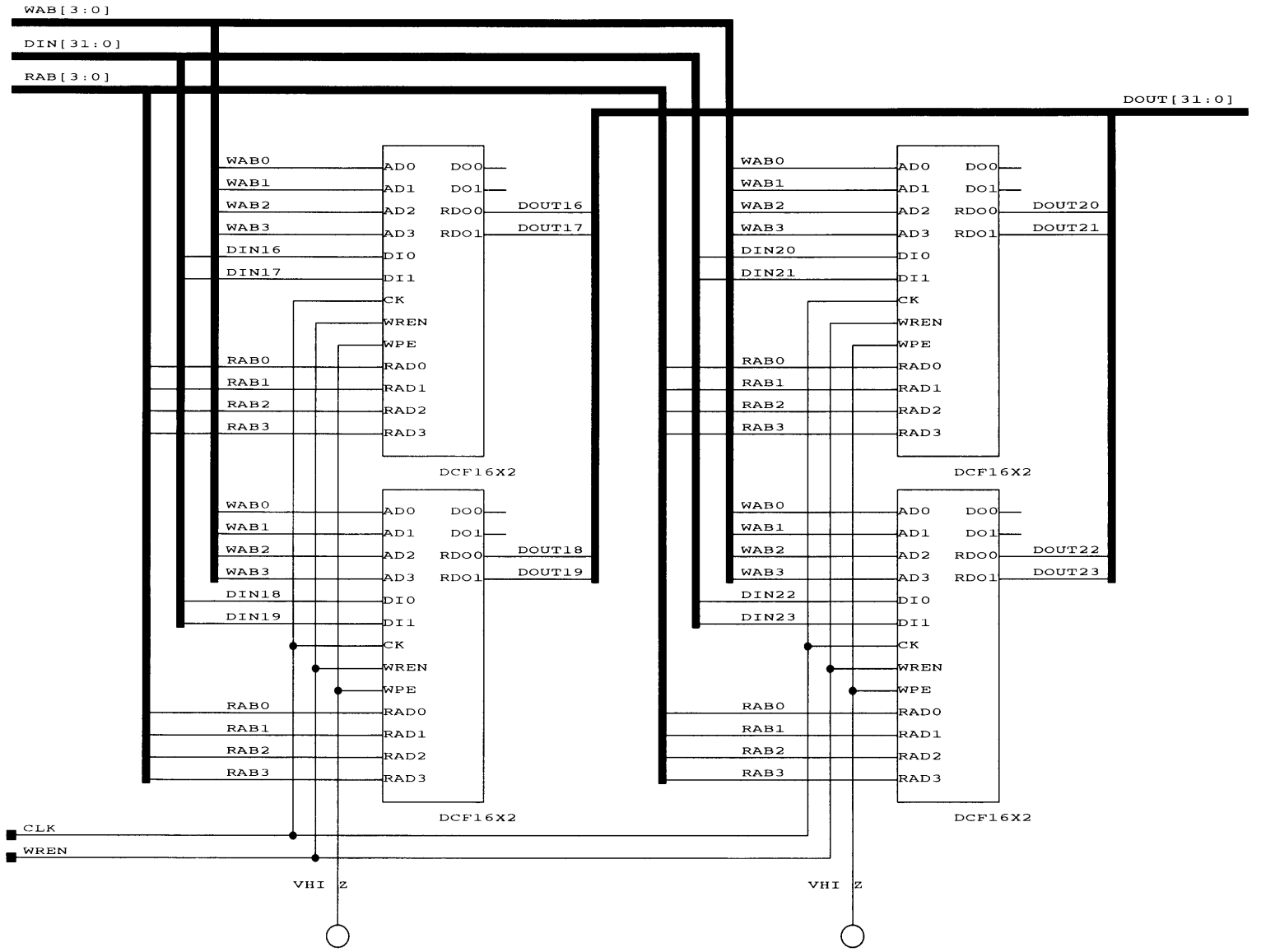

| | | |
|---|---|---|
| dde | Dansk Data Elektronik A/S | |
| Issue 0 | 970901 | BSP301-1<br>PCI BE INPUT REGISTER<br>& BE SWOPPER |
| Issue 1 | | |
| Issue 2 | | |
| Issue 3 | | File: pcibein Page:1 of 1 |



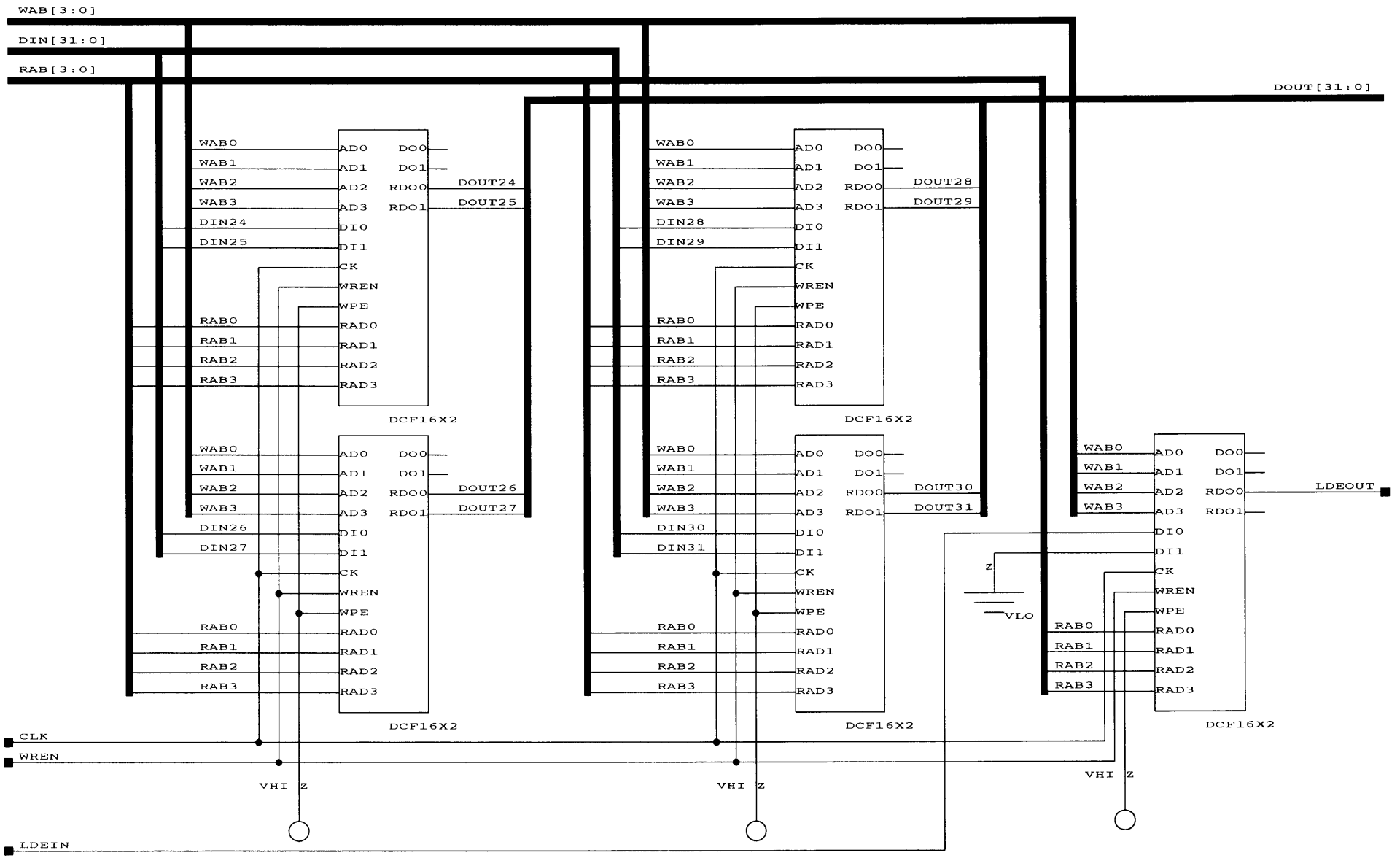

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | DATA BUFFER - 16 WORDS |
| Issue 2 | | |
| Issue 3 | | File: pcidbuf Page:1 of 5 |



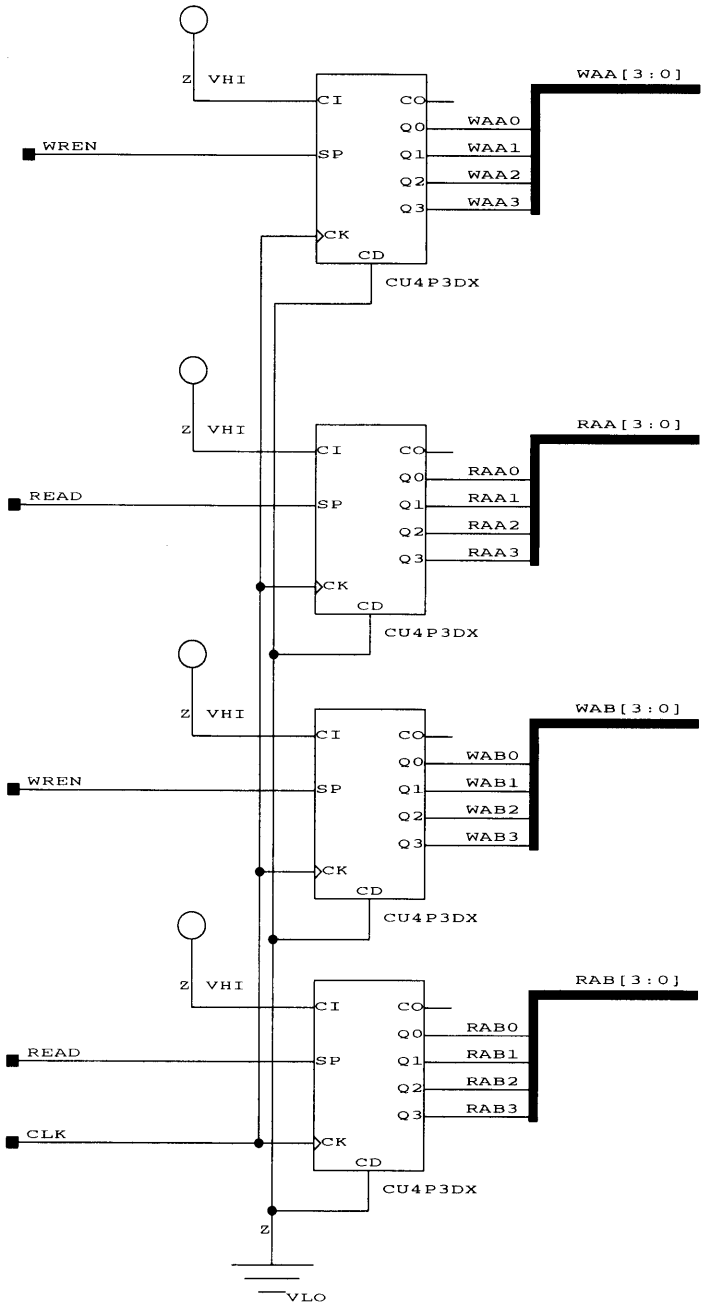

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>DATA BUFFER - 16 WORDS | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: pcidbuf | Page:2 of 5 |

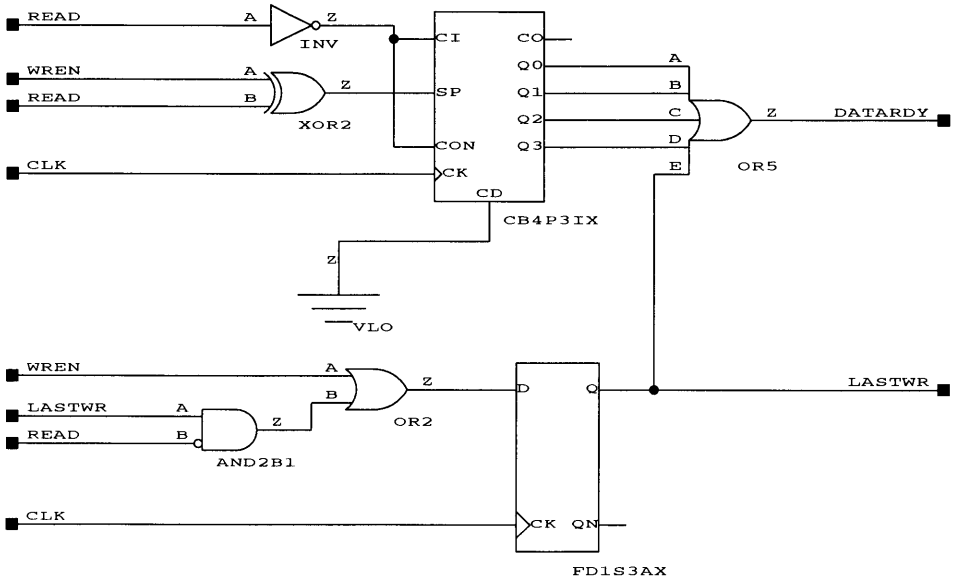



| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | DATA BUFFER - 16 WORDS |
| Issue 2 | | |
| Issue 3 | | File: pcidbuf Page:3 of 5 |

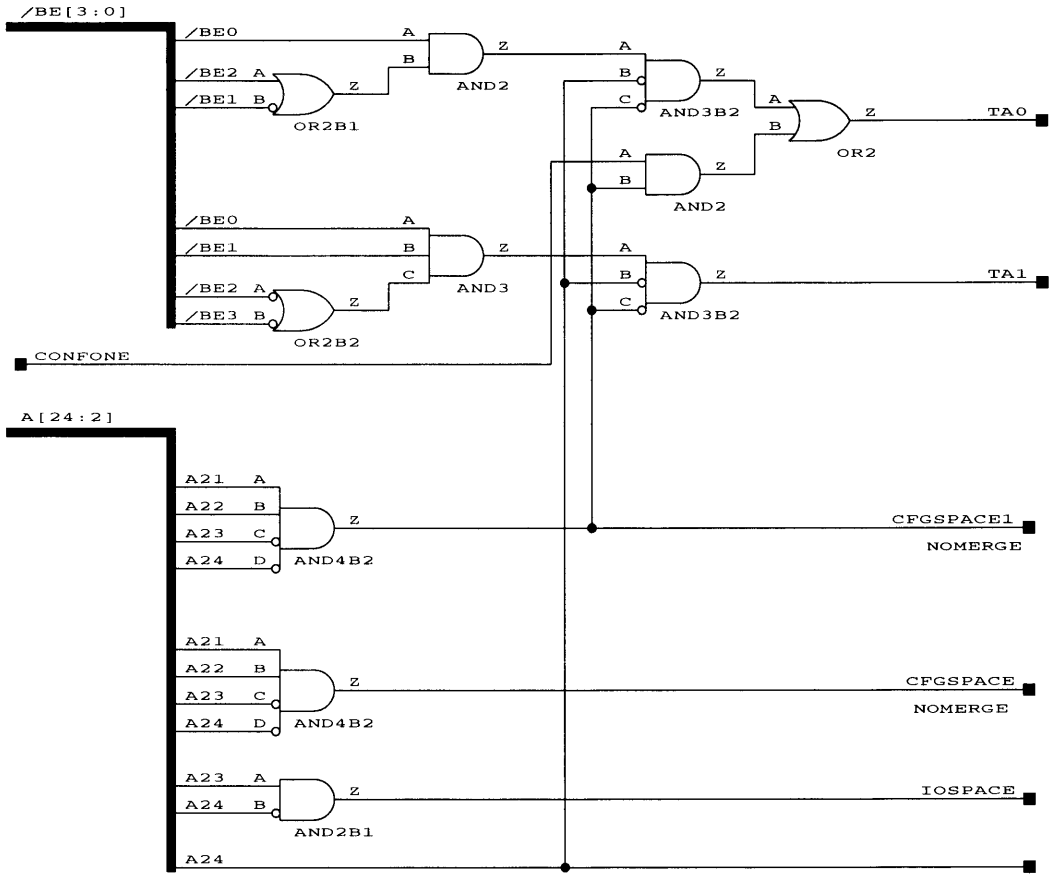



| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 | |
| Issue 1 | | DATA BUFFER - 16 WORDS | |
| Issue 2 | | | |
| Issue 3 | | File: pcidbuf | Page:4 of 5 |

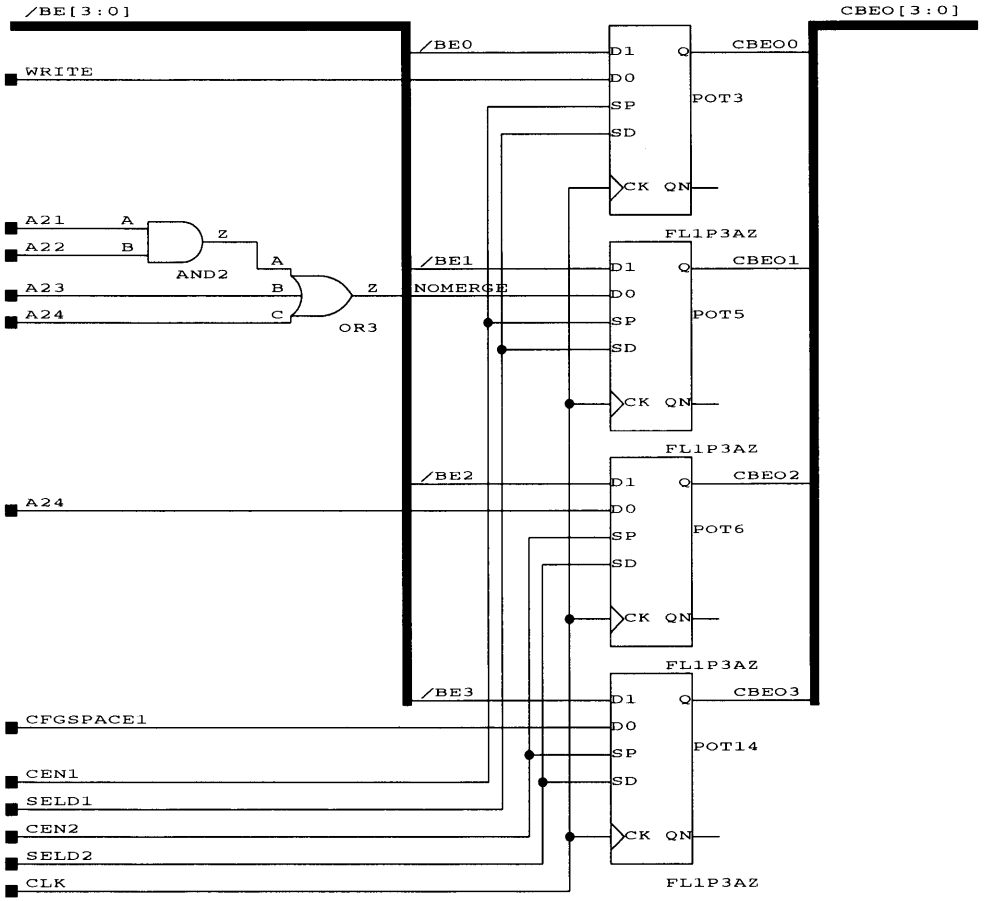


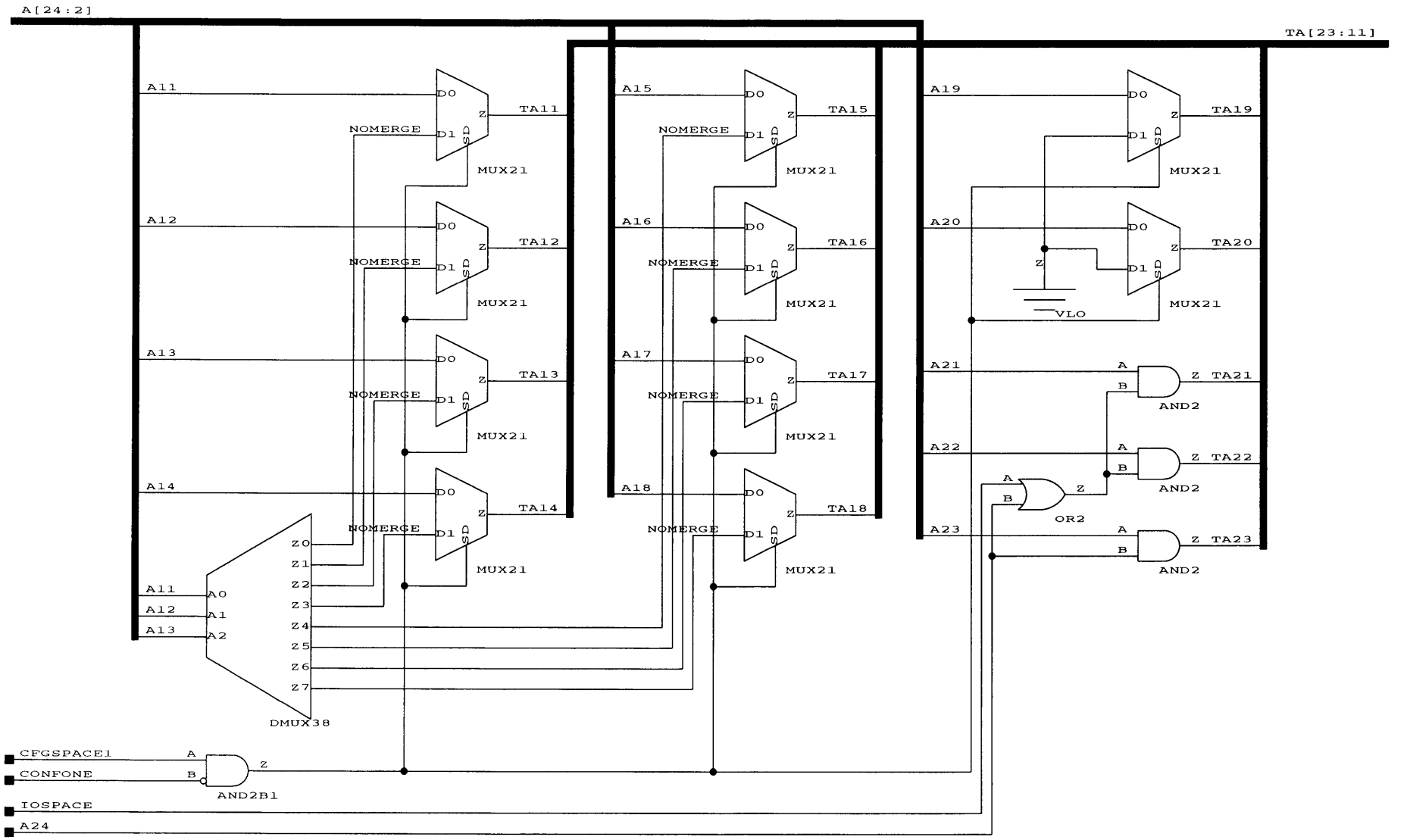

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | DATA BUFFER - 16 WORDS |
| Issue 2 | | |
| Issue 3 | | File: pcidbuf Page:5 of 5 |



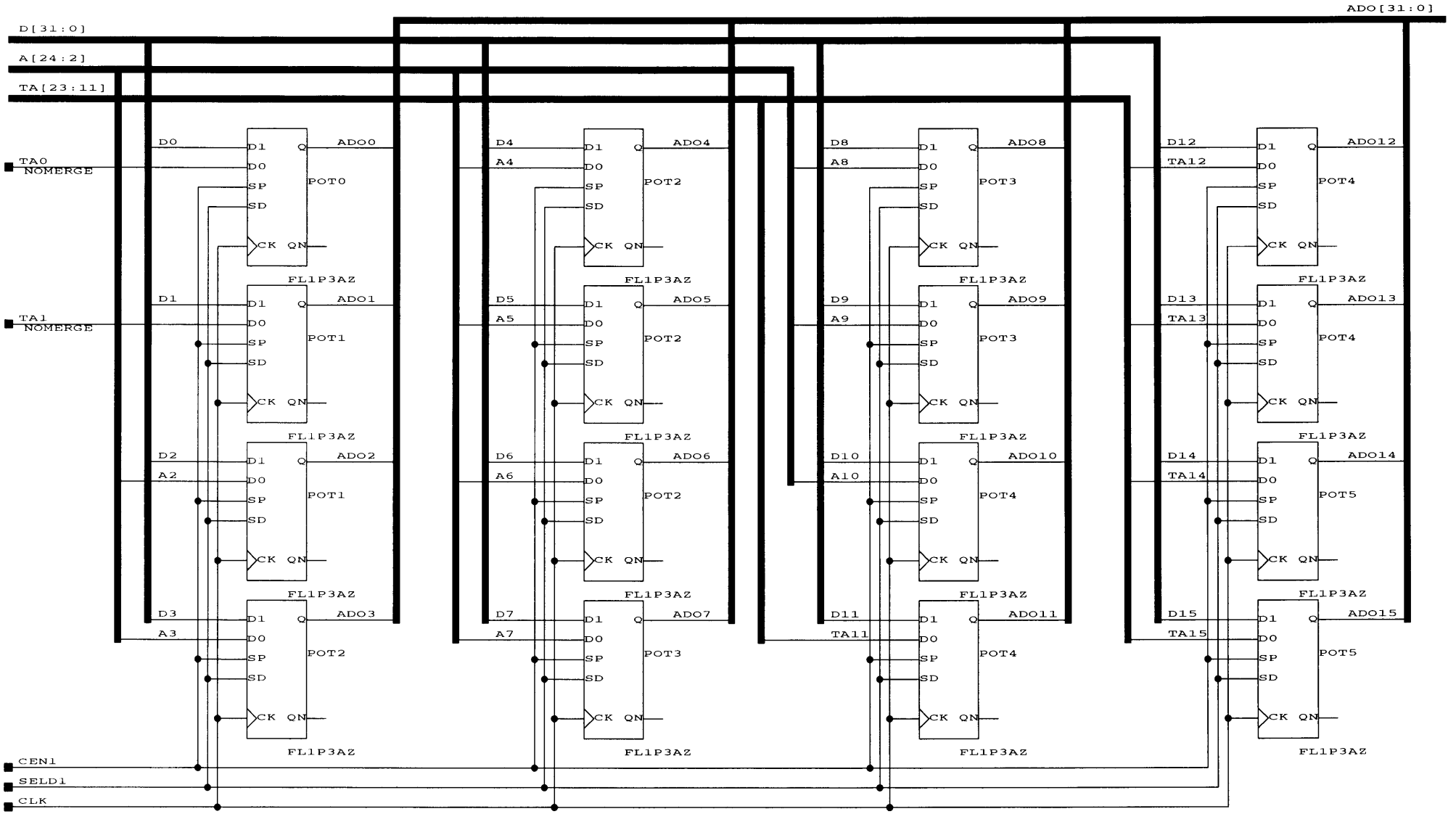

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI ADDR/DATA OUTPUT MULTIPLEXER & REGISTER |
| Issue 2 | | |
| Issue 3 | | File: pciadmux Page:1 of 5 |



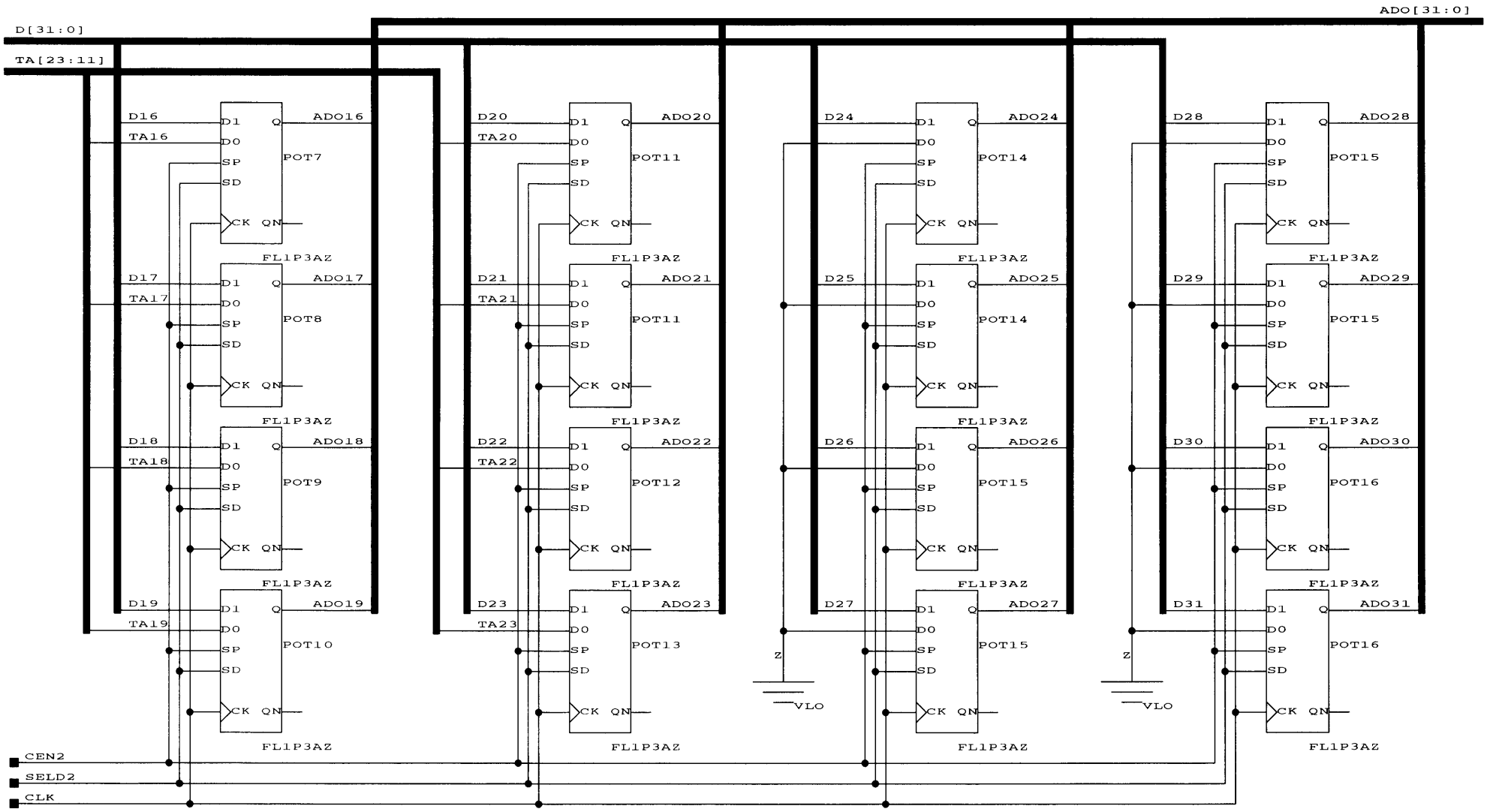

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>PCI ADDR/DATA OUTPUT<br>MULTIPLEXER & REGISTER |
| Issue 1 | | |
| Issue 2 | | |
| Issue 3 | | File: pciadmux Page:2 of 5 |



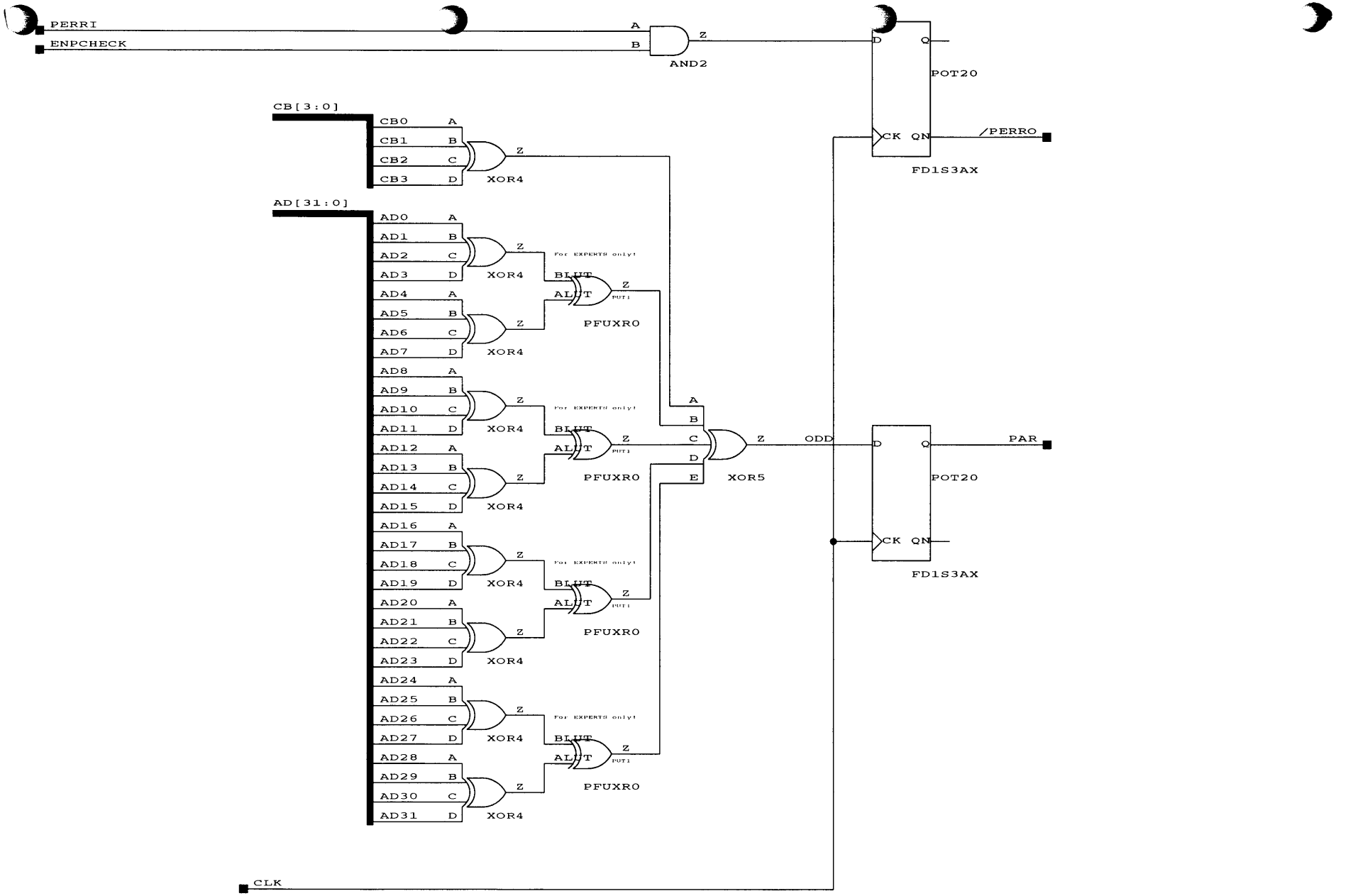

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI ADDR/DATA OUTPUT |
| Issue 2 | | MULTIPLEXER & REGISTER |
| Issue 3 | | File: pciadmux Page:3 of 5 |



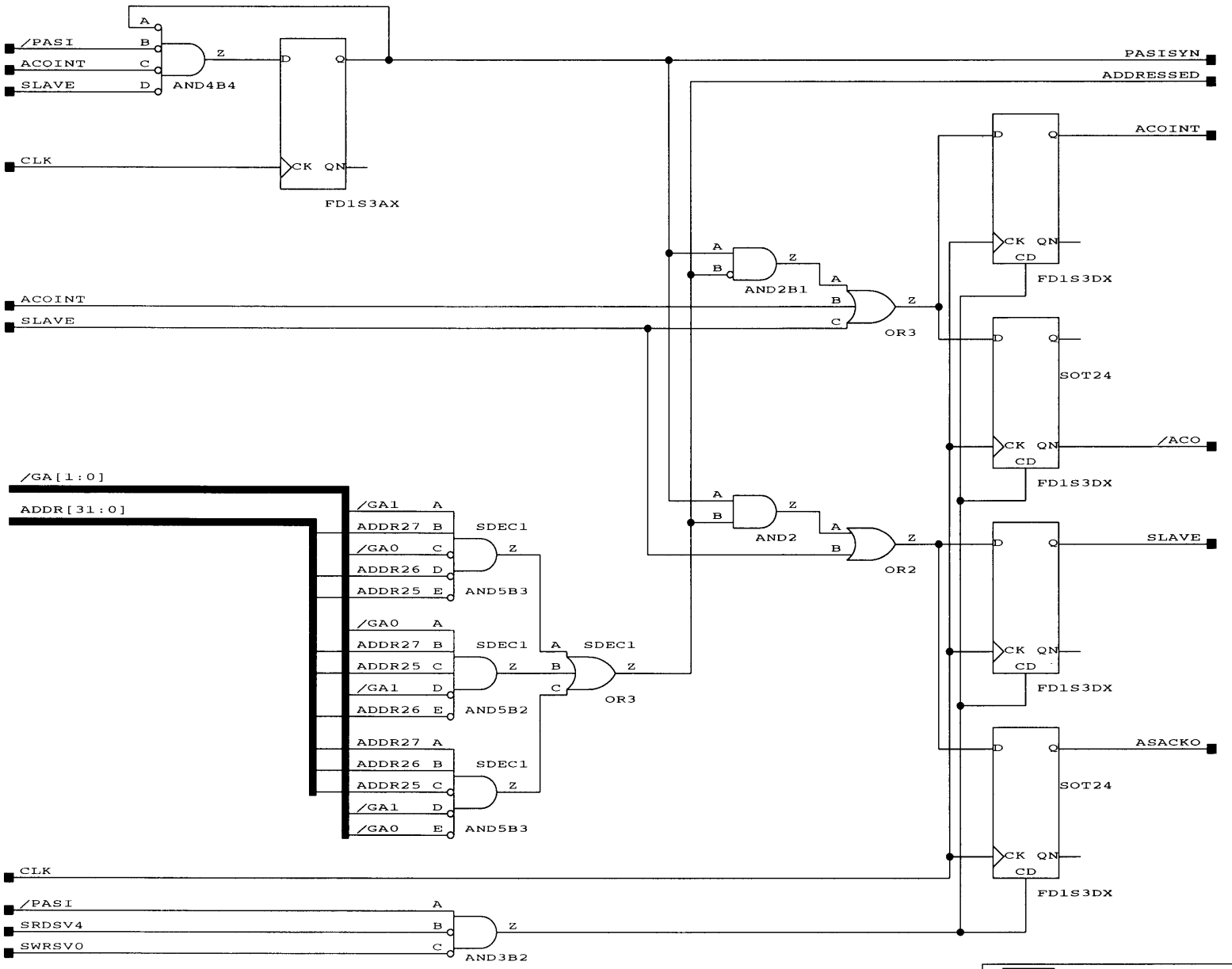

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI ADDR/DATA OUTPUT |
| Issue 2 | | MULTIPLEXER & REGISTER |
| Issue 3 | | File: pciadmux Page:4 of 5 |



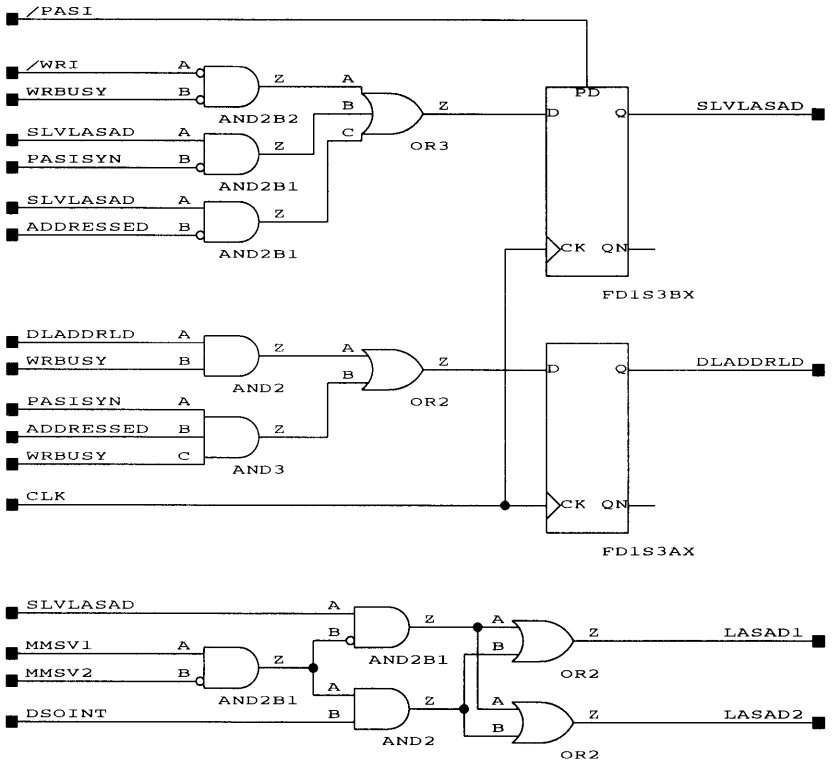

| | | |
|---|---|---|
| dde | Dansk Data Elektronik A/S | |
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI ADDR/DATA OUTPUT MULTIPLEXER & REGISTER |
| Issue 2 | | |
| Issue 3 | | File: pciadmux Page:5 of 5 |

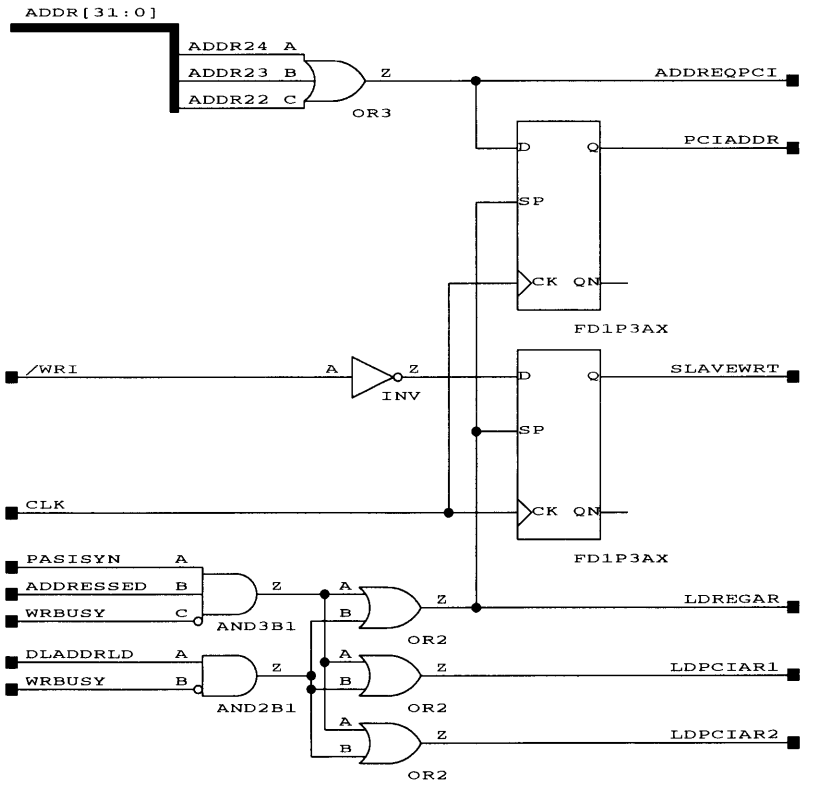



| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970917 | BSP301-1 | |
| Issue 1 | | PCI PARITY GENERATE | |
| Issue 2 | | | |
| Issue 3 | | File: pcipgen | Page:1 of 1 |

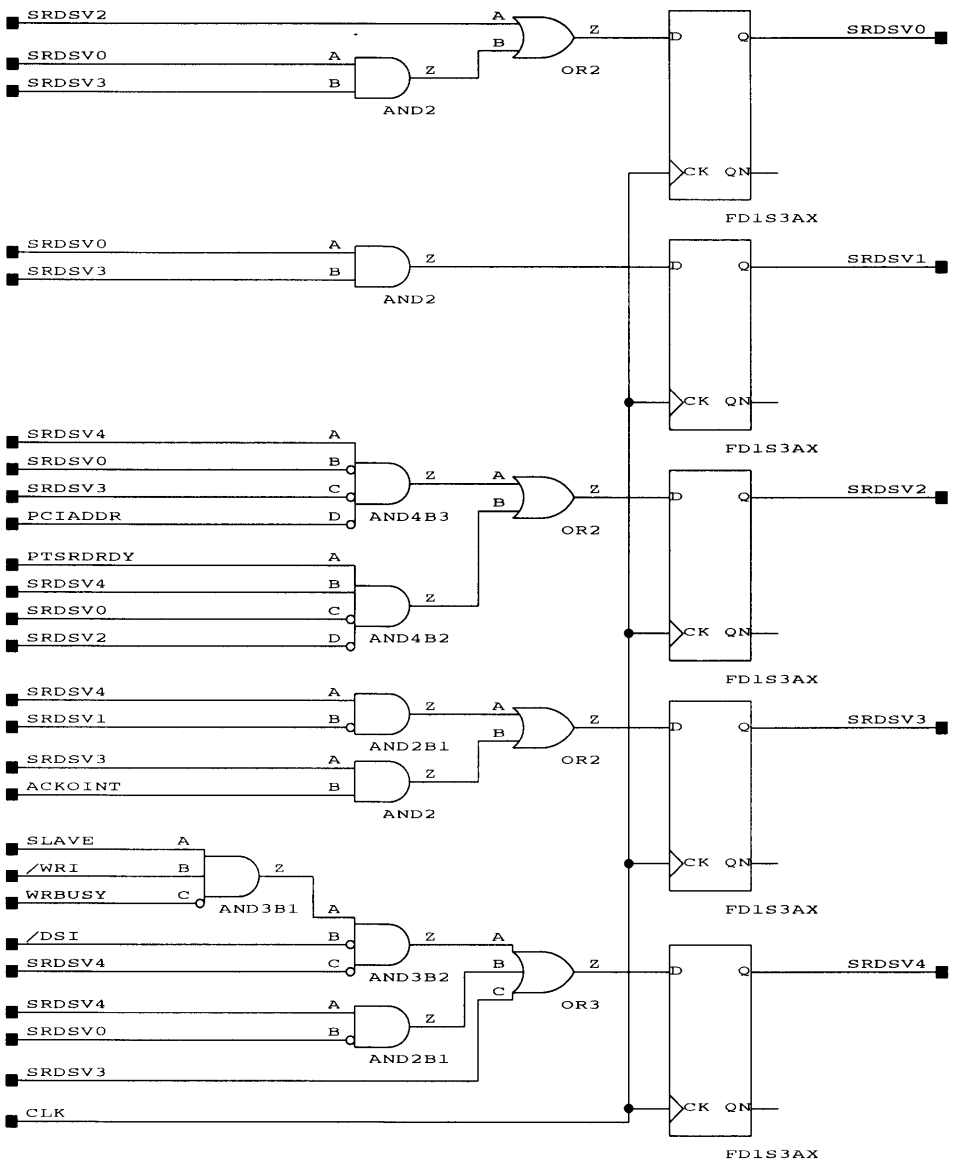


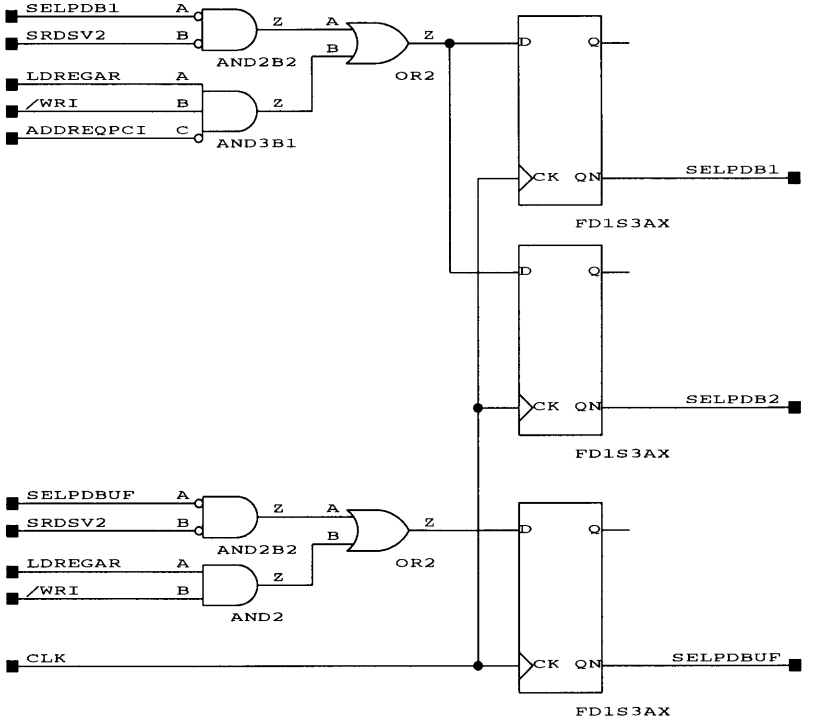

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI MASTER/SLAVE CONTROL |
| Issue 2 | | Slave Address Control |
| Issue 3 | | File: smimsctr Page:1 of 12 |



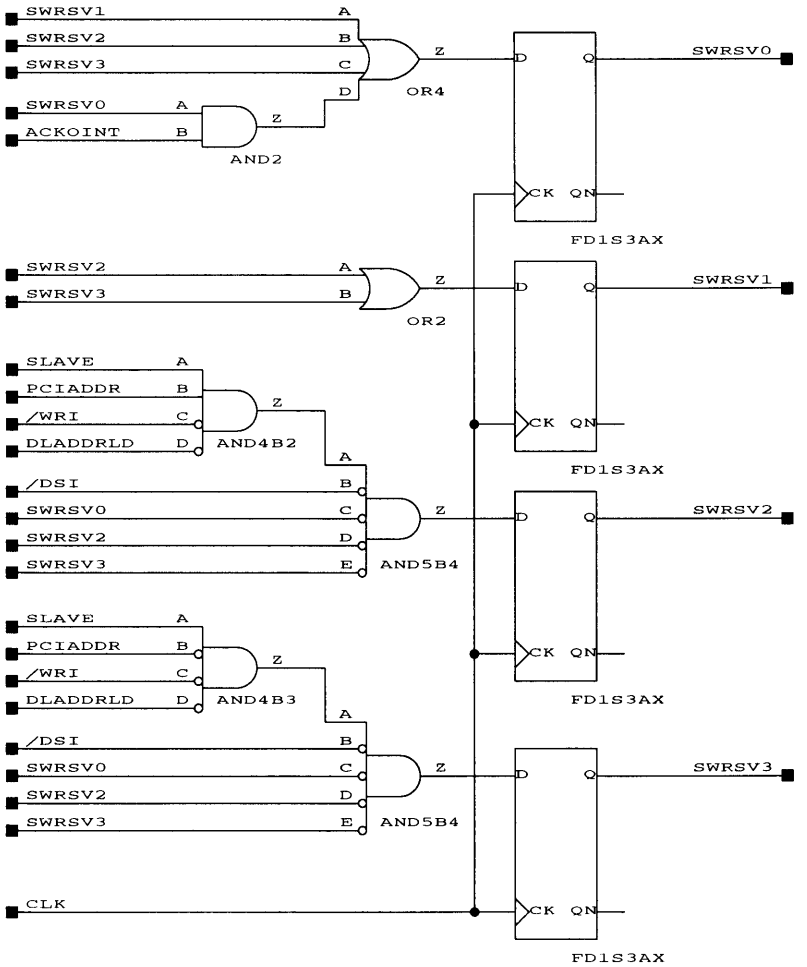




| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI MASTER/SLAVE CONTROL |
| Issue 2 | | Slave Address Control |
| Issue 3 | | File: smimsctr Page:2 of 12 |

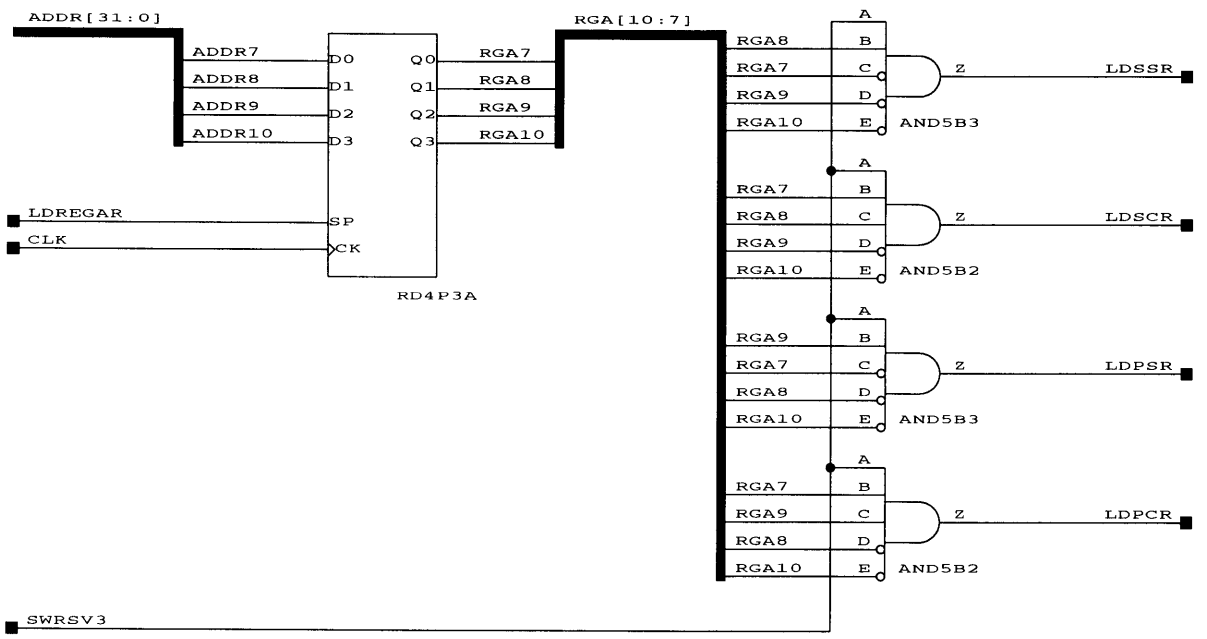


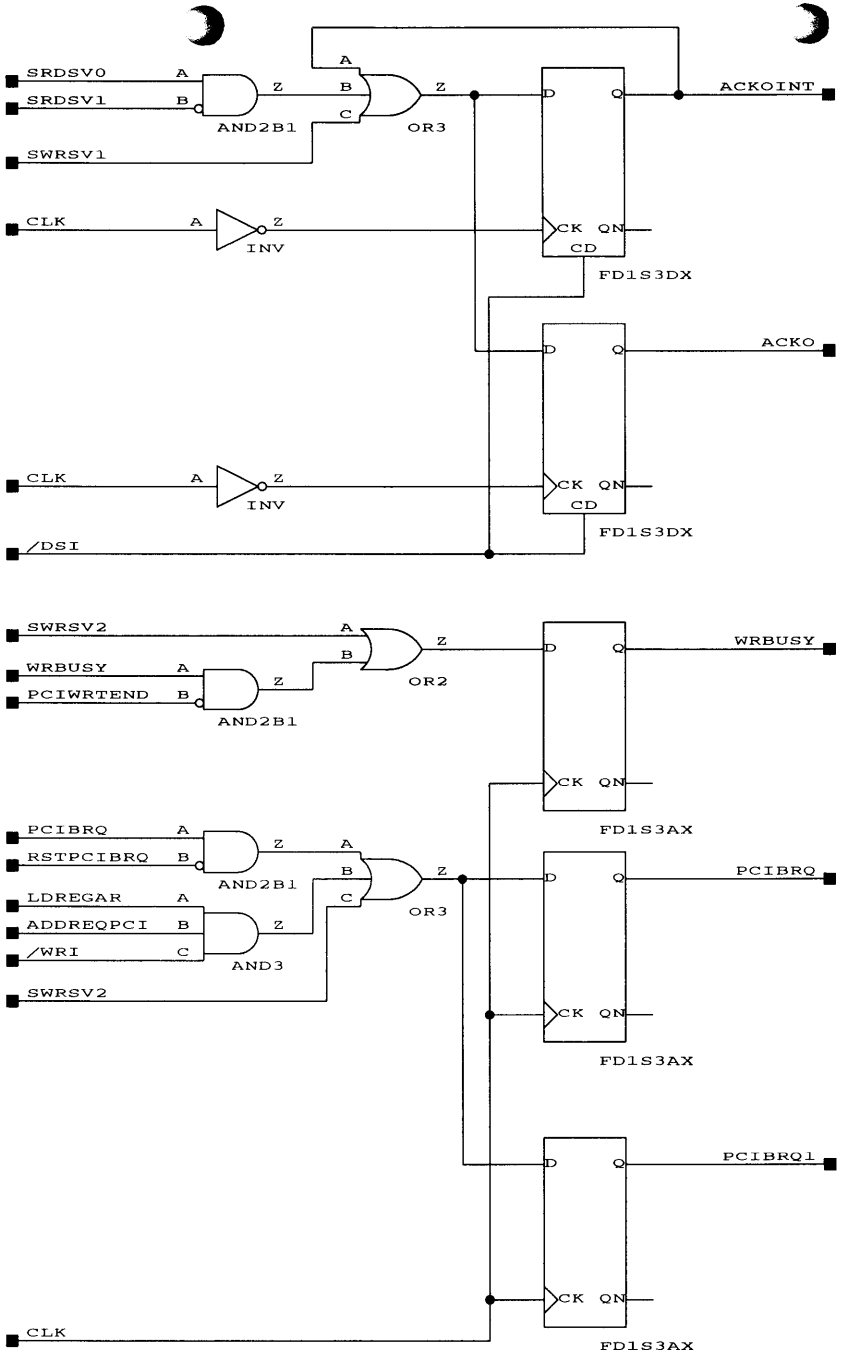

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI MASTER/SLAVE CONTROL |
| Issue 2 | | Slave Read Control |
| Issue 3 | | File: smimsctr Page:3 of 12 |



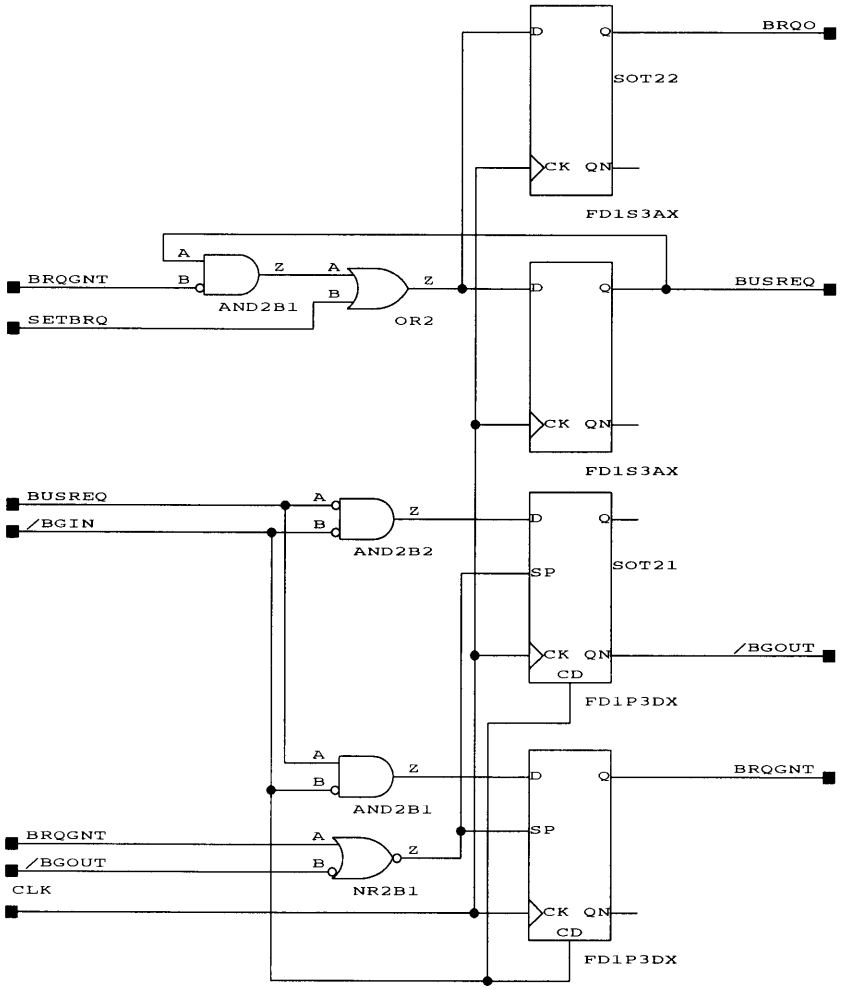

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>SMI MASTER/SLAVE CONTROL<br>Slave Write Control | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: smimsctr | Page:4 of 12 |



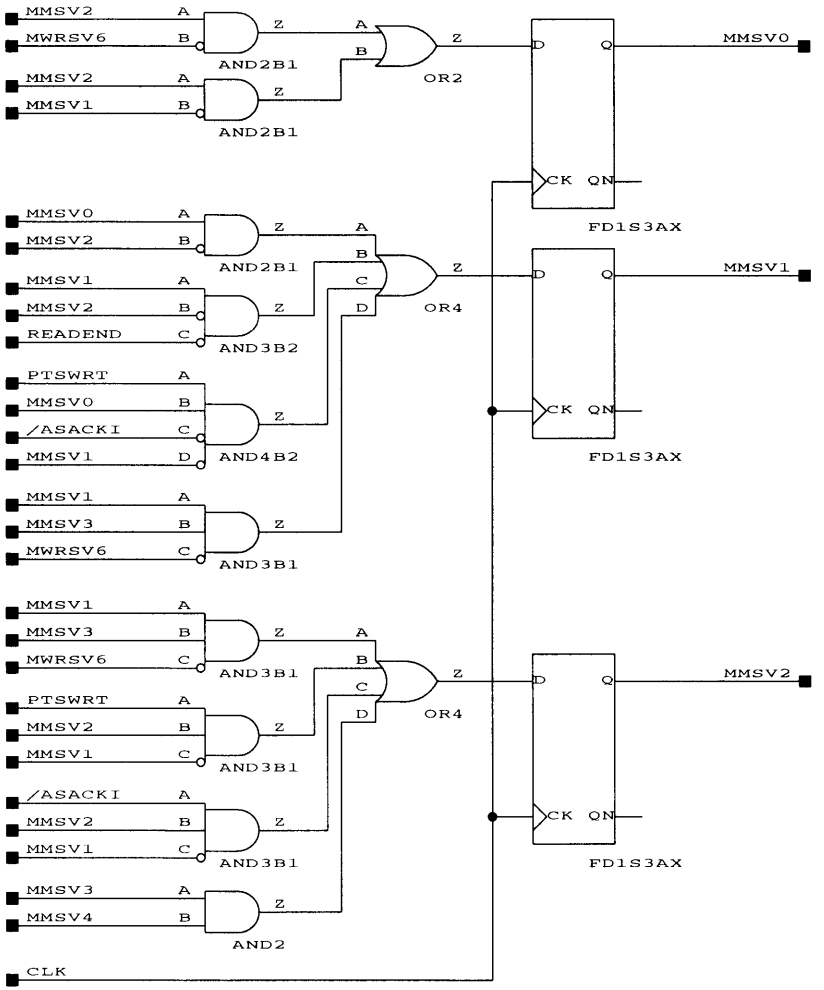

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 | |
| Issue 1 | | SMI MASTER/SLAVE CONTROL | |
| Issue 2 | | Slave Write Control | |
| Issue 3 | | File: smimsctr | Page:5 of 12 |

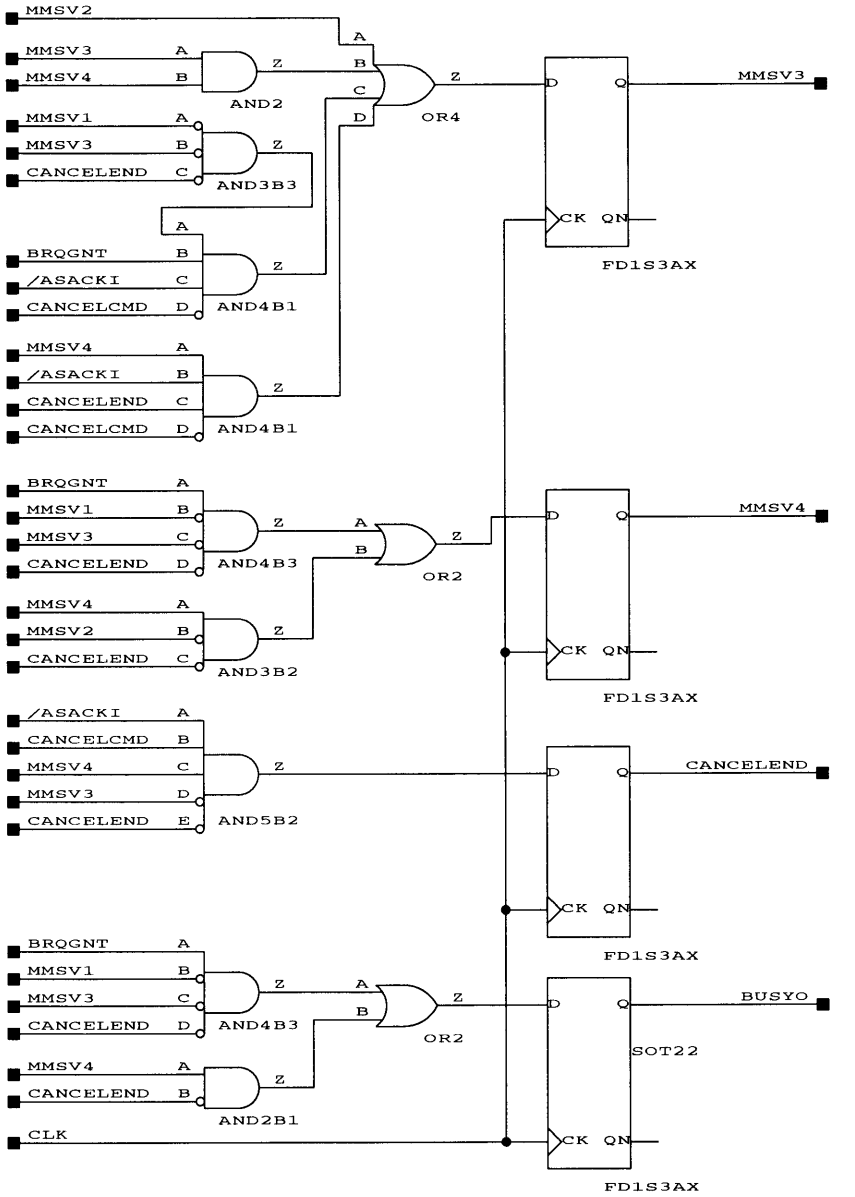



| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI MASTER/SLAVE CONTROL |
| Issue 2 | | Slave Read/Write Control |
| Issue 3 | | File: smimsctr Page:6 of 12 |

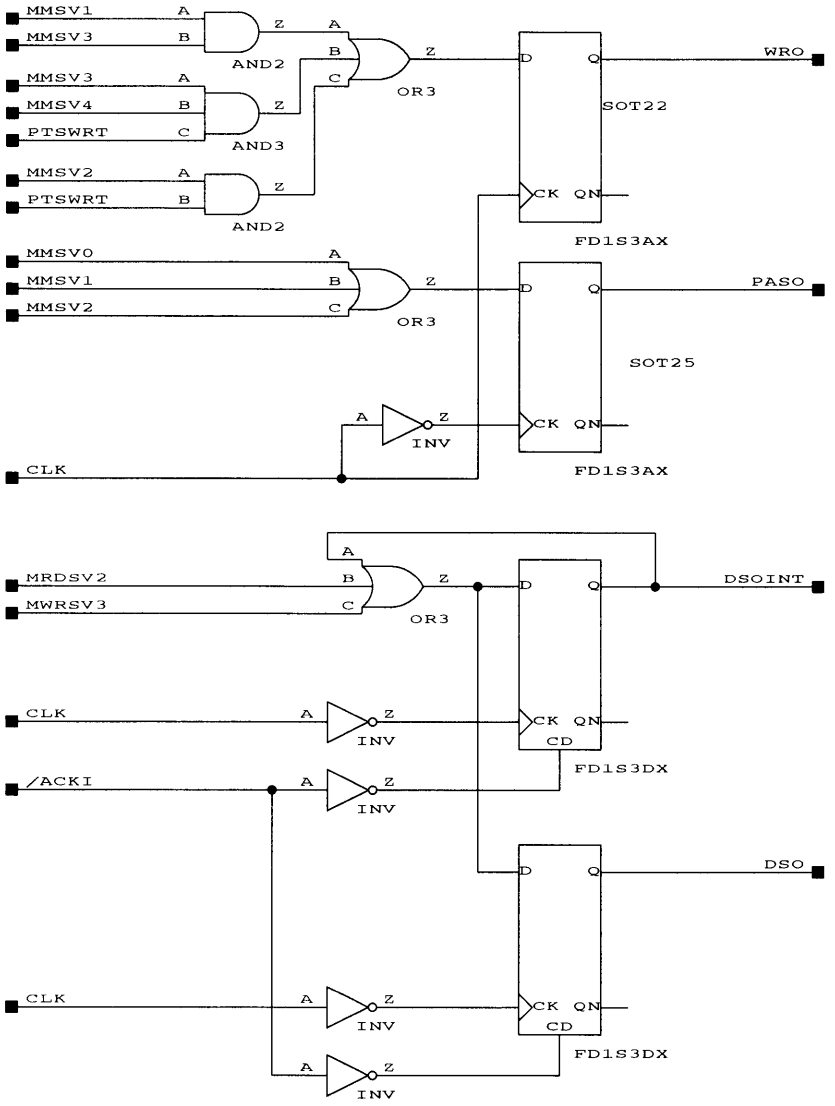


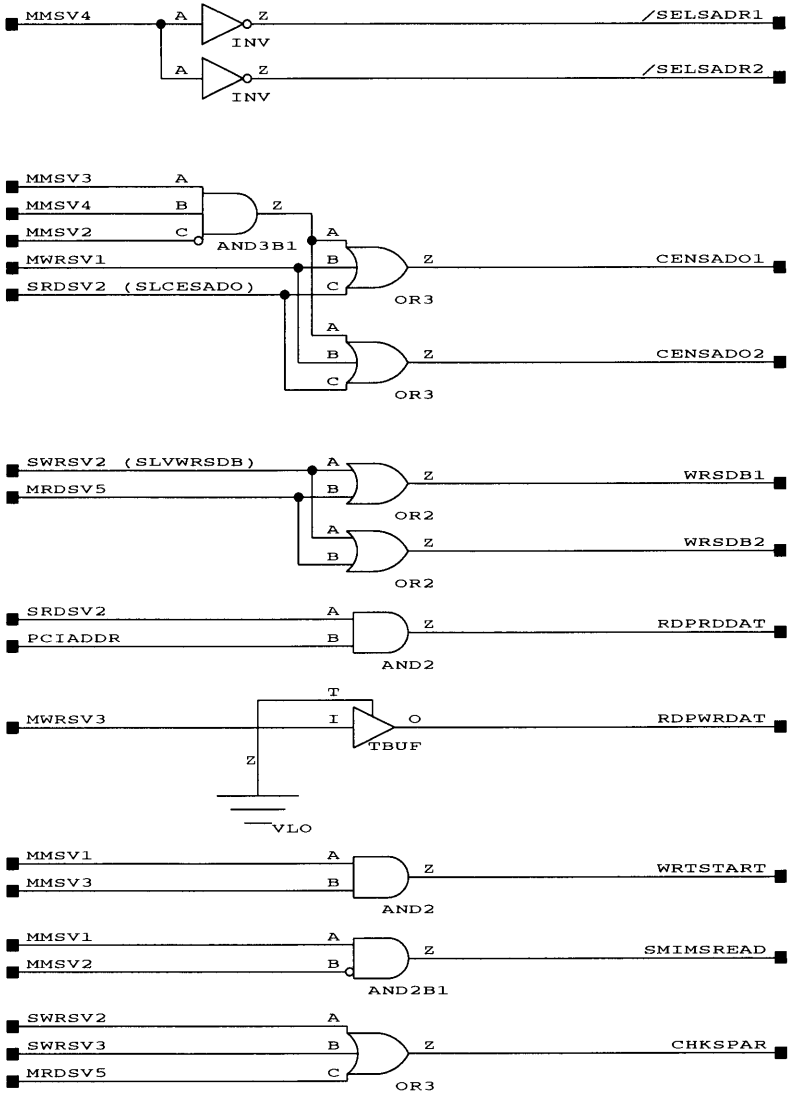

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 | |
| Issue 1 | | SMI MASTER/SLAVE CONTROL | |
| Issue 2 | | Bus Request & Abitration | |
| Issue 3 | | File: smimsctr | Page:7 of 12 |



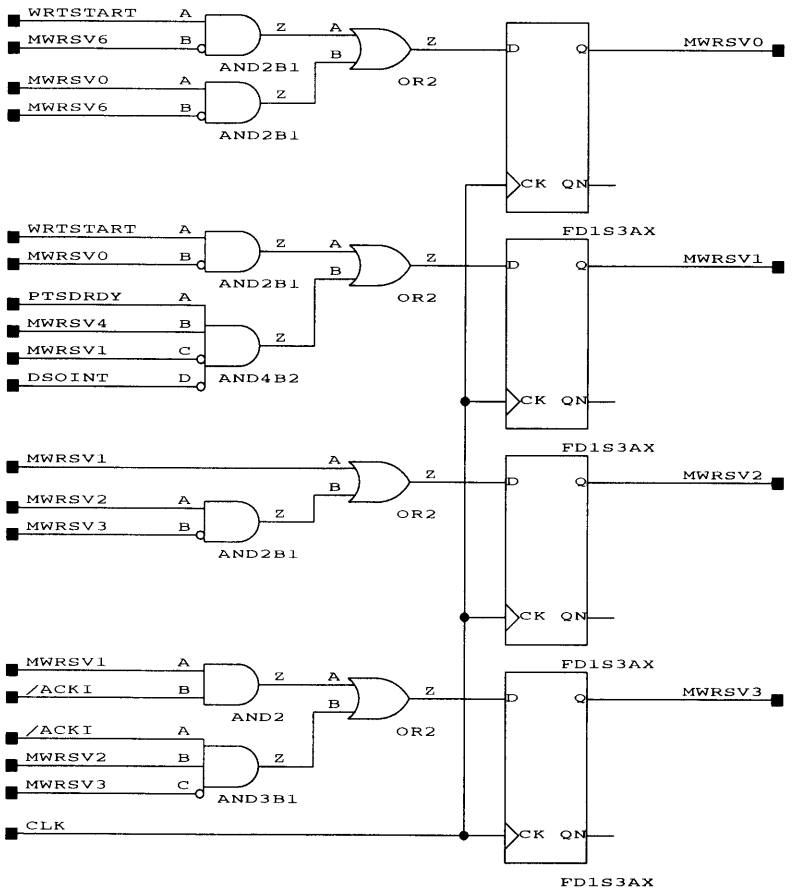



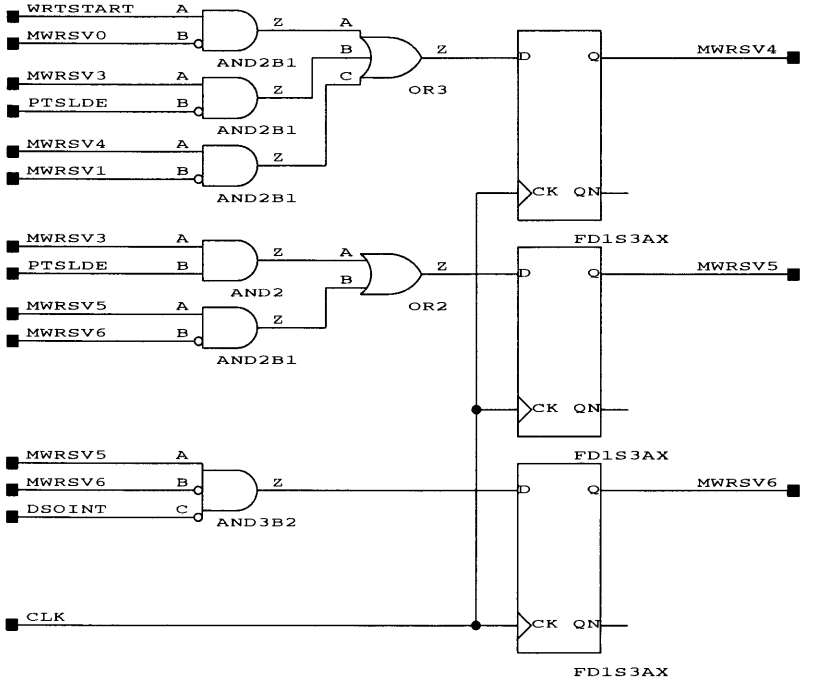

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI MASTER/SLAVE CONTROL |
| Issue 2 | | Master Main State Machine |
| Issue 3 | | File: smimsctr Page:8 of 12 |



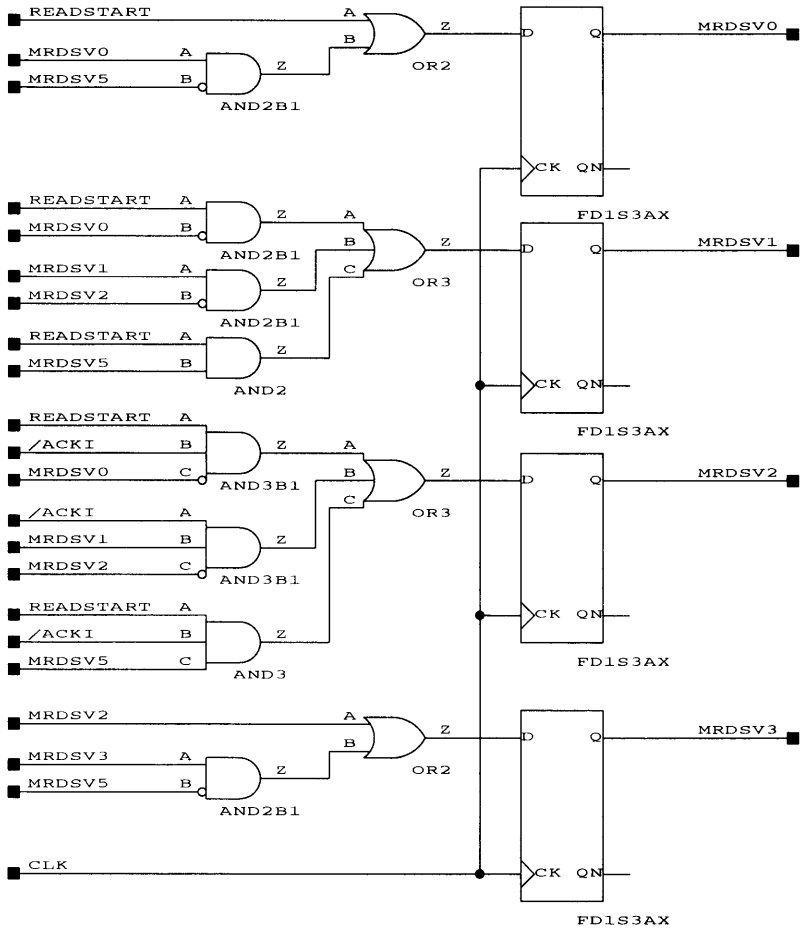



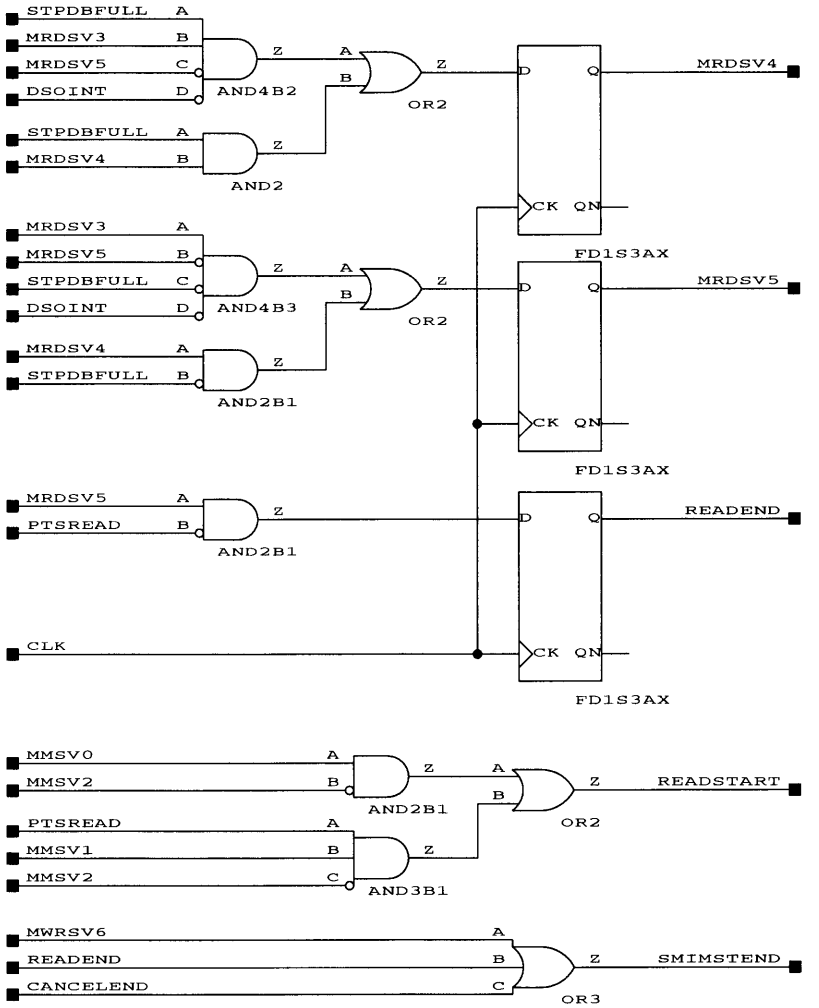

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI MASTER/SLAVE CONTROL |
| Issue 2 | | Master Main State Machine |
| Issue 3 | | File: smimsctr Page:9 of 12 |



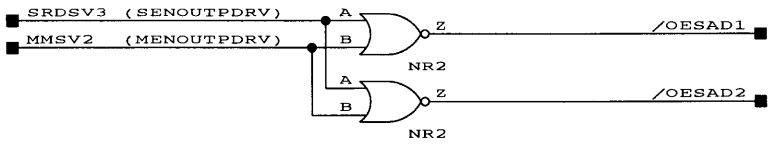



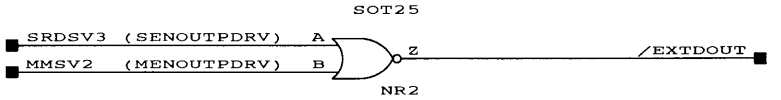

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 | |
| Issue 1 | | SMI MASTER/SLAVE CONTROL | |
| Issue 2 | | Master Write State Machine | |
| Issue 3 | | File: smimsctr | Page:10 of 12 |

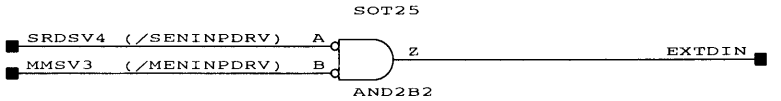


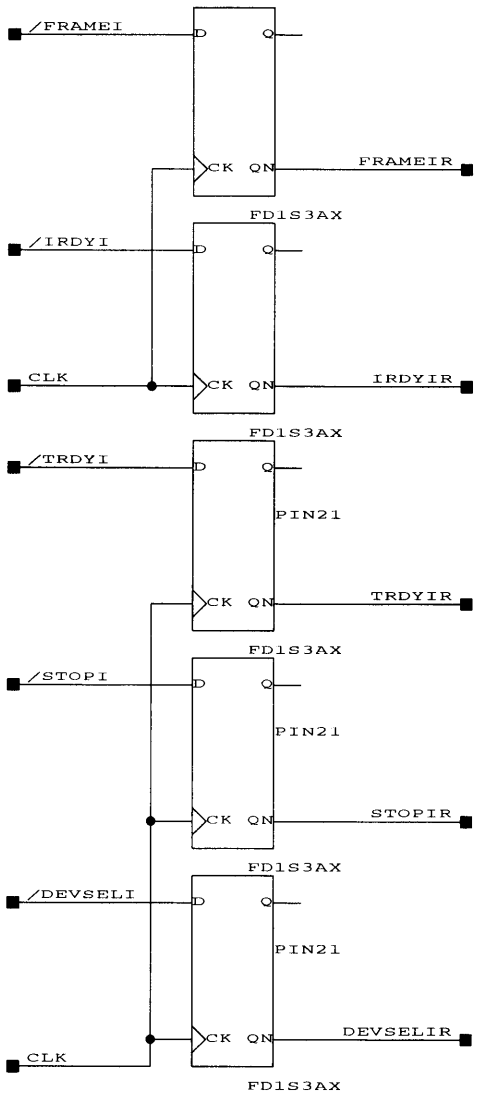

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | SMI MASTER/SLAVE CONTROL |
| Issue 2 | | Master Read State Machine |
| Issue 3 | | File: smimsctr Page:11 of 12 |

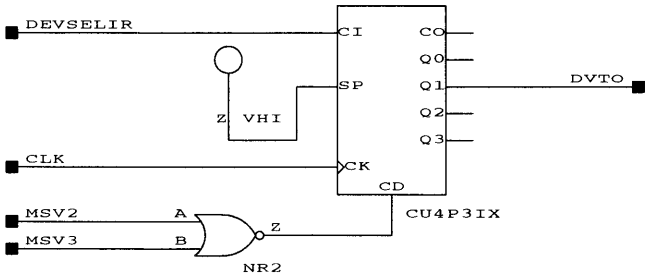


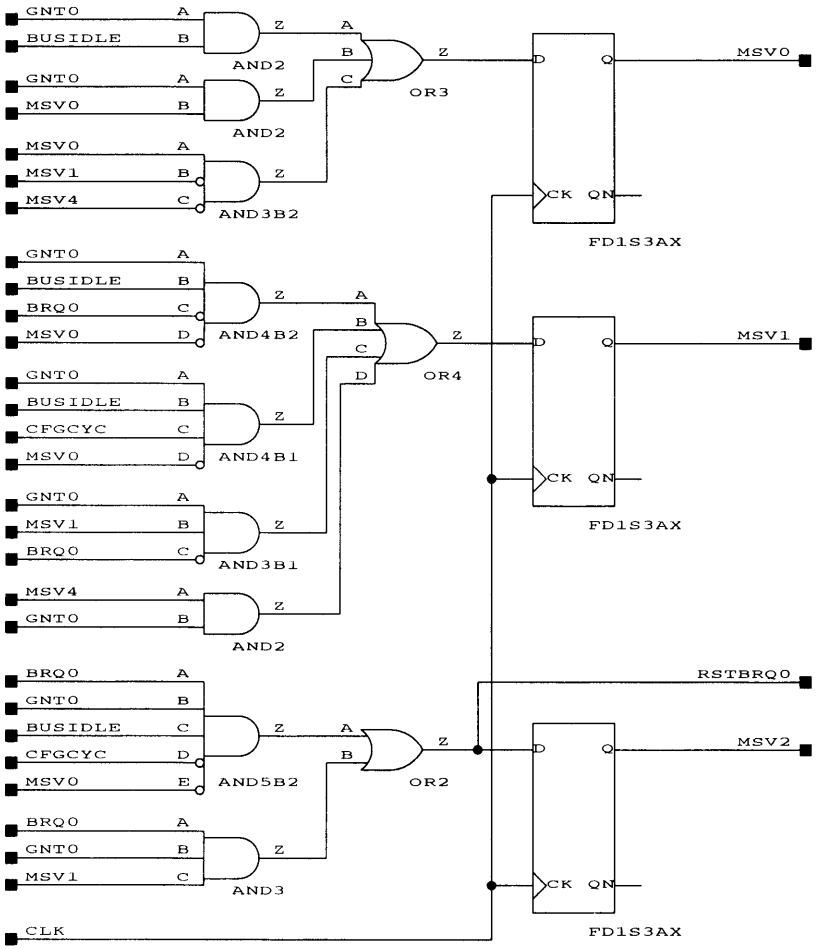



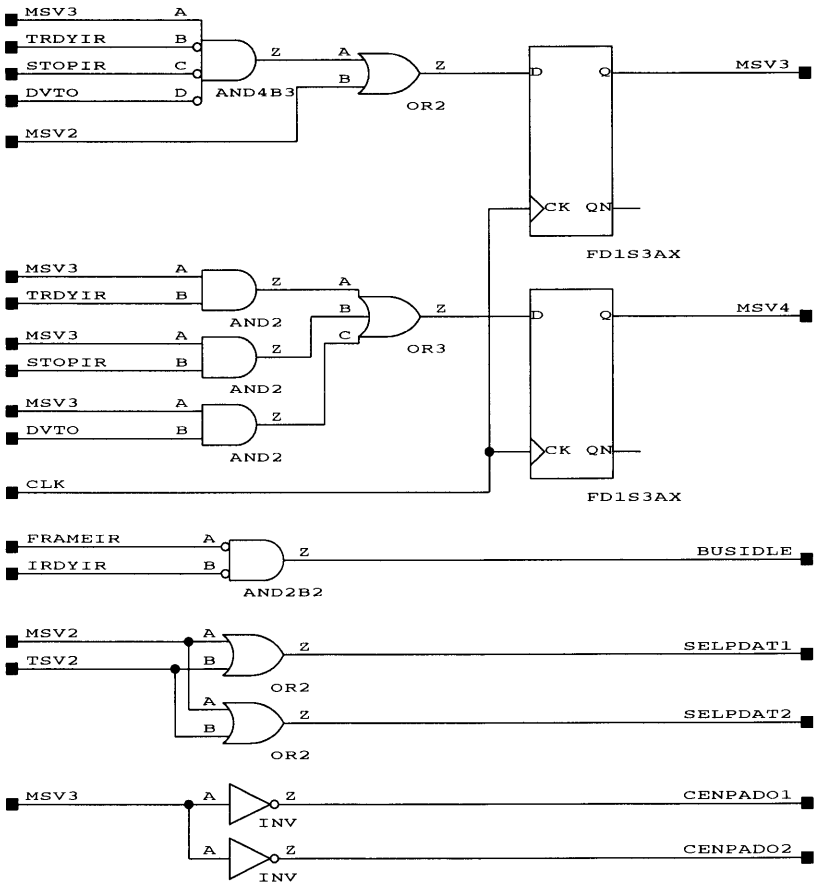

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1<br>SMI MASTER/SLAVE CONTROL<br>Transceiver Control | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: smimsctr | Page:12 of 12 |



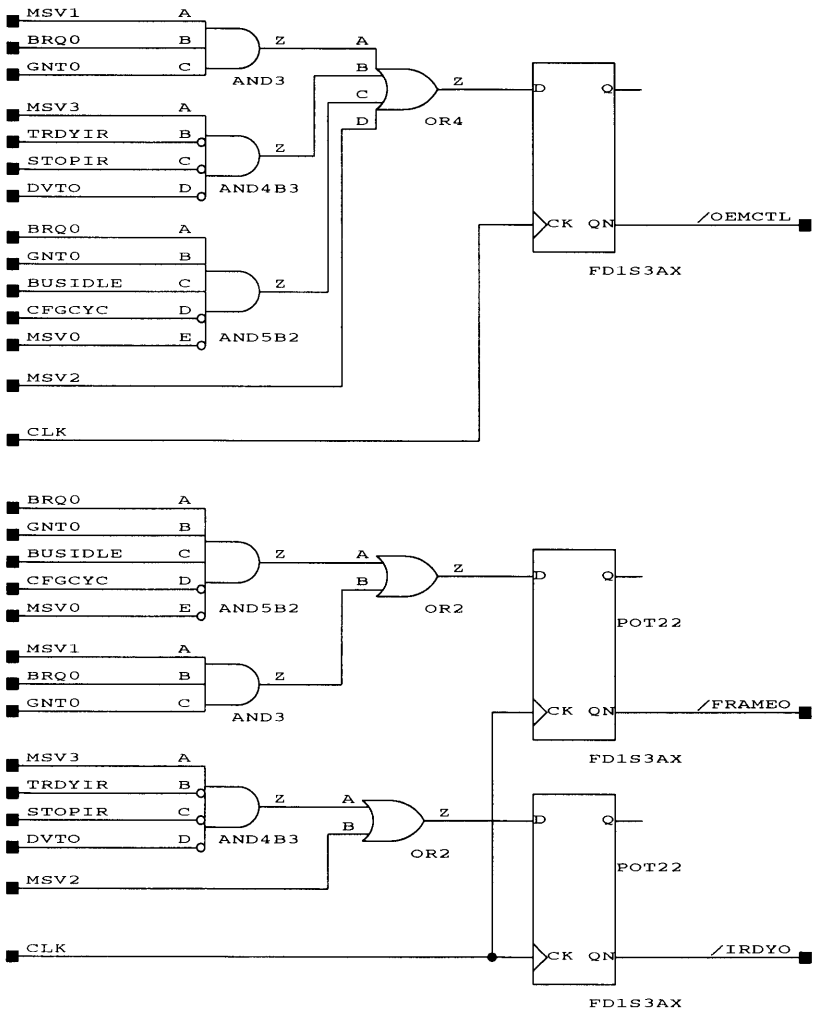



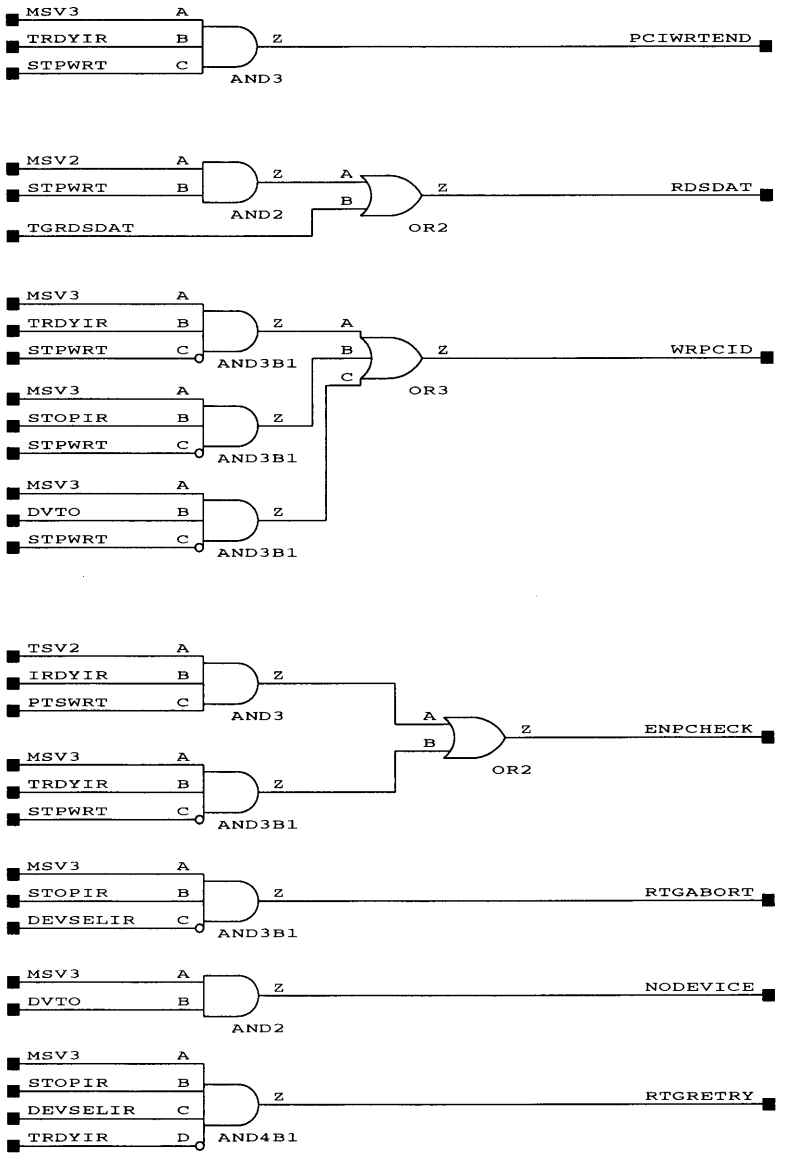

| | | |
|---|---|---|
| dde | Dansk Data Elektronik A/S | |
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI MASTER/TARGET CONTROL |
| Issue 2 | | Master Control Logic |
| Issue 3 | | File: pcimtctr Page:1 of 11 |



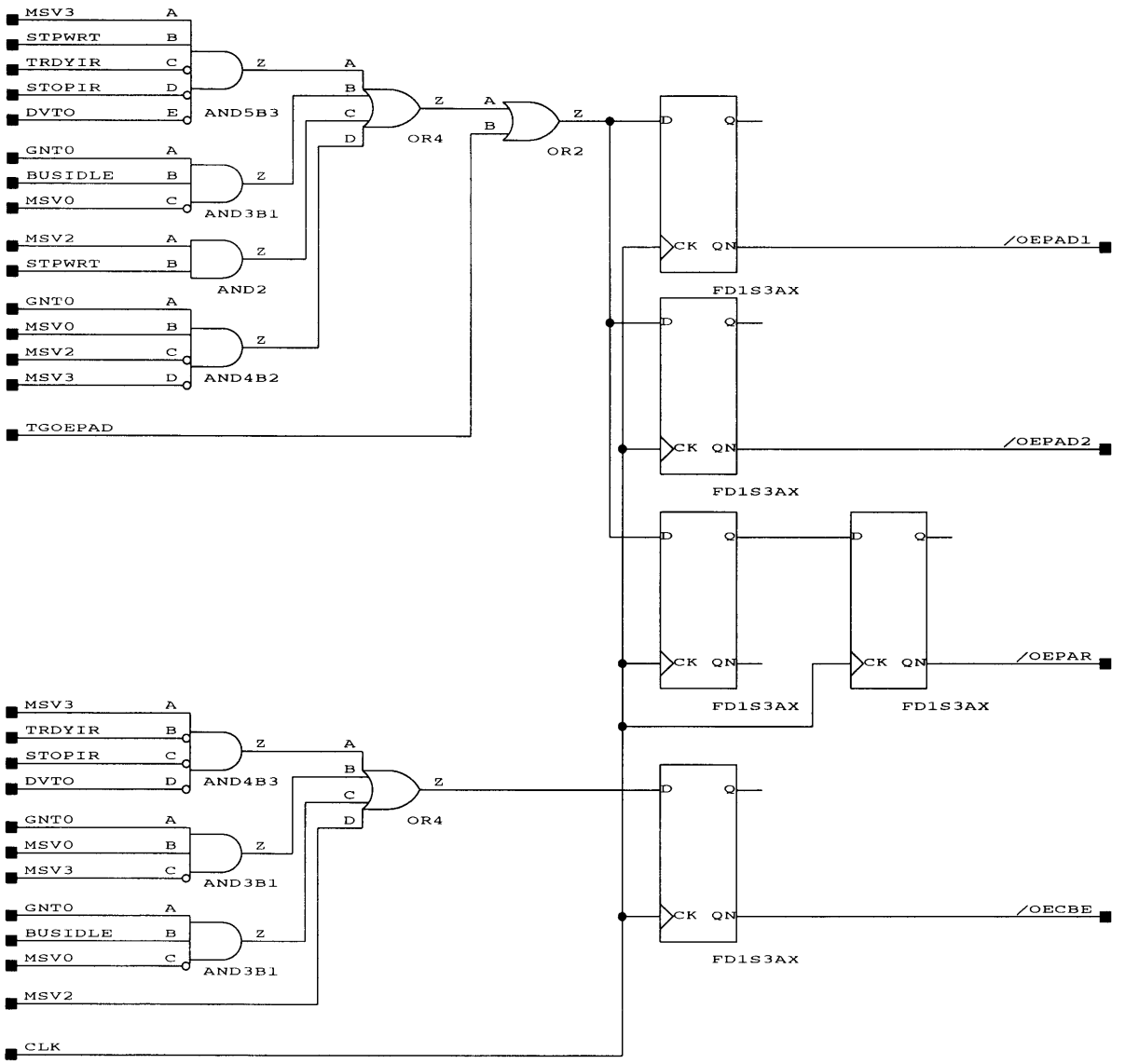




| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI MASTER/TARGET CONTROL |
| Issue 2 | | Master Control Logic |
| Issue 3 | | File: pcimtctr Page: 2 of 11 |

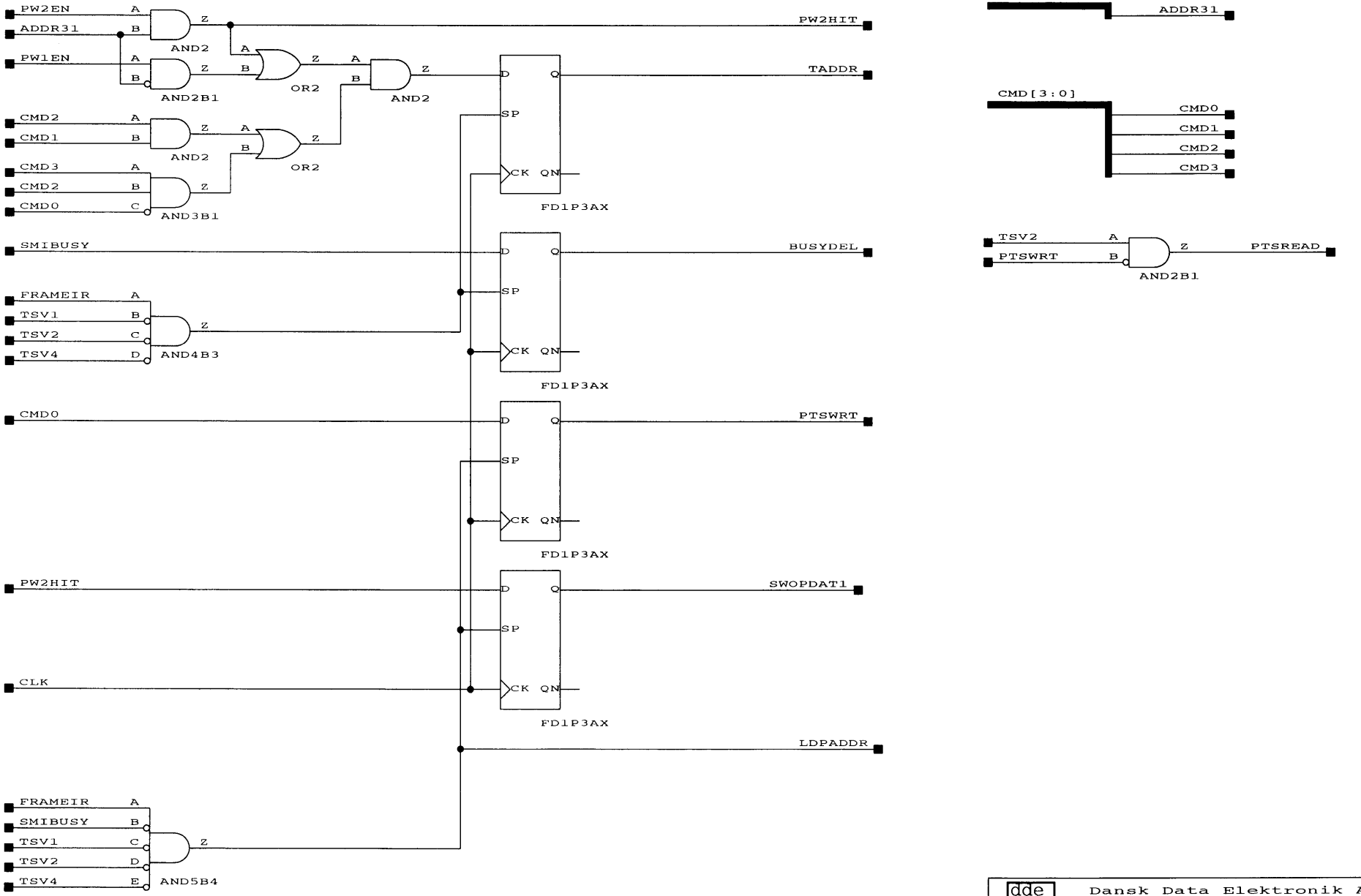


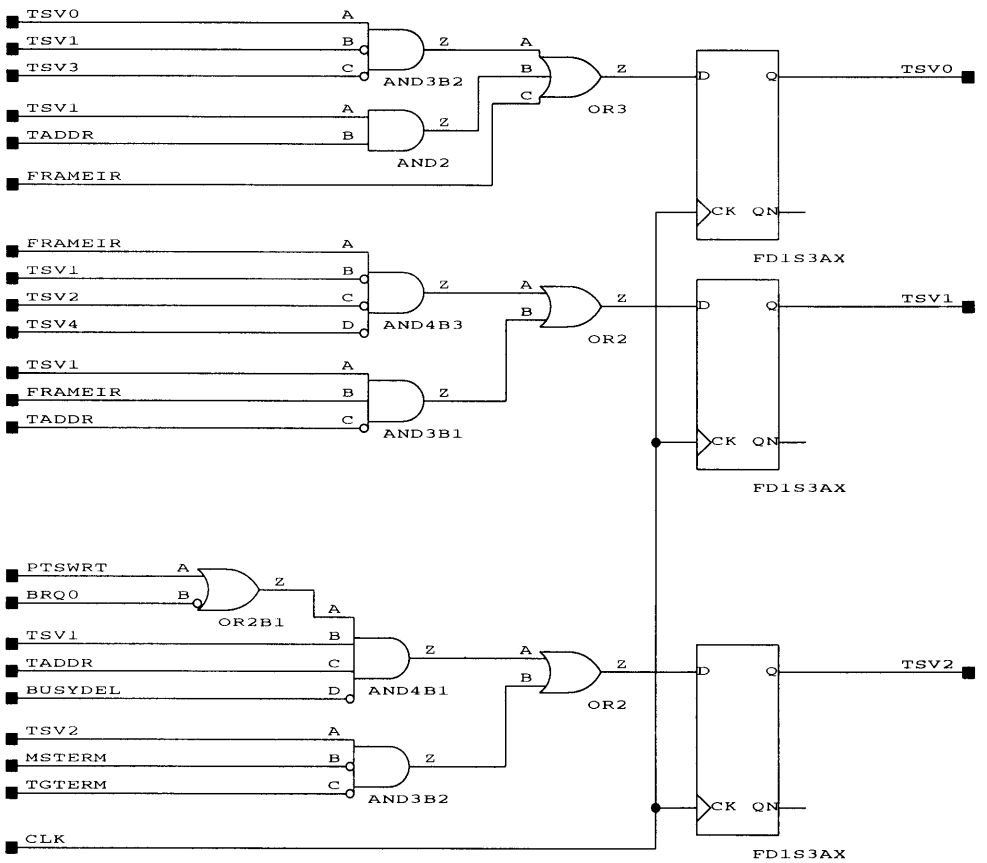

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970917 | BSP301-1 |
| Issue 1 | | PCI MASTER/TARGET CONTROL |
| Issue 2 | | Master Control Logic |
| Issue 3 | | File: pcimtctr Page:3 of 11 |

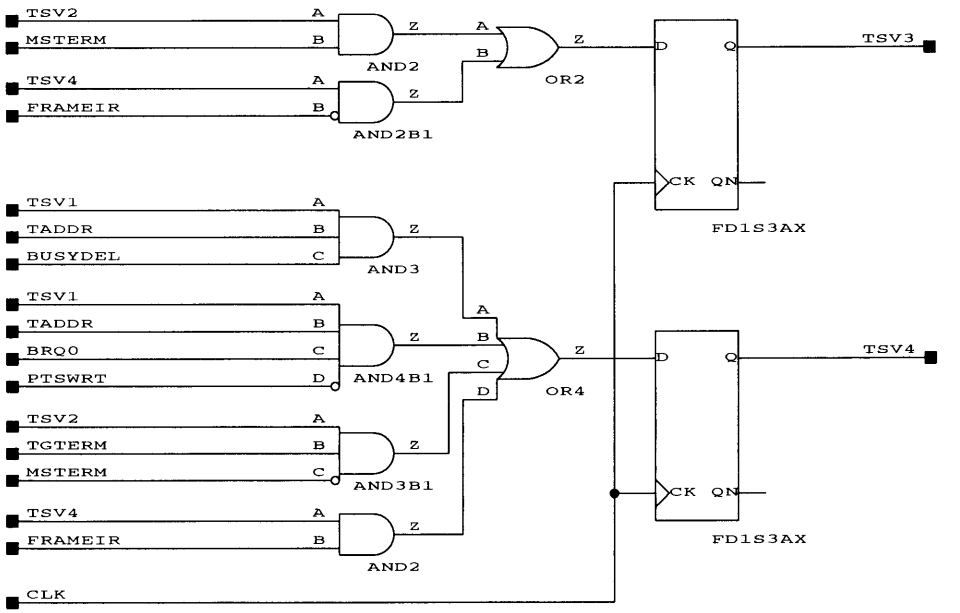



| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI MASTER/TARGET CONTROL |
| Issue 2 | | Master Control Logic |
| Issue 3 | | File: pcimtctr Page:4 of 11 |

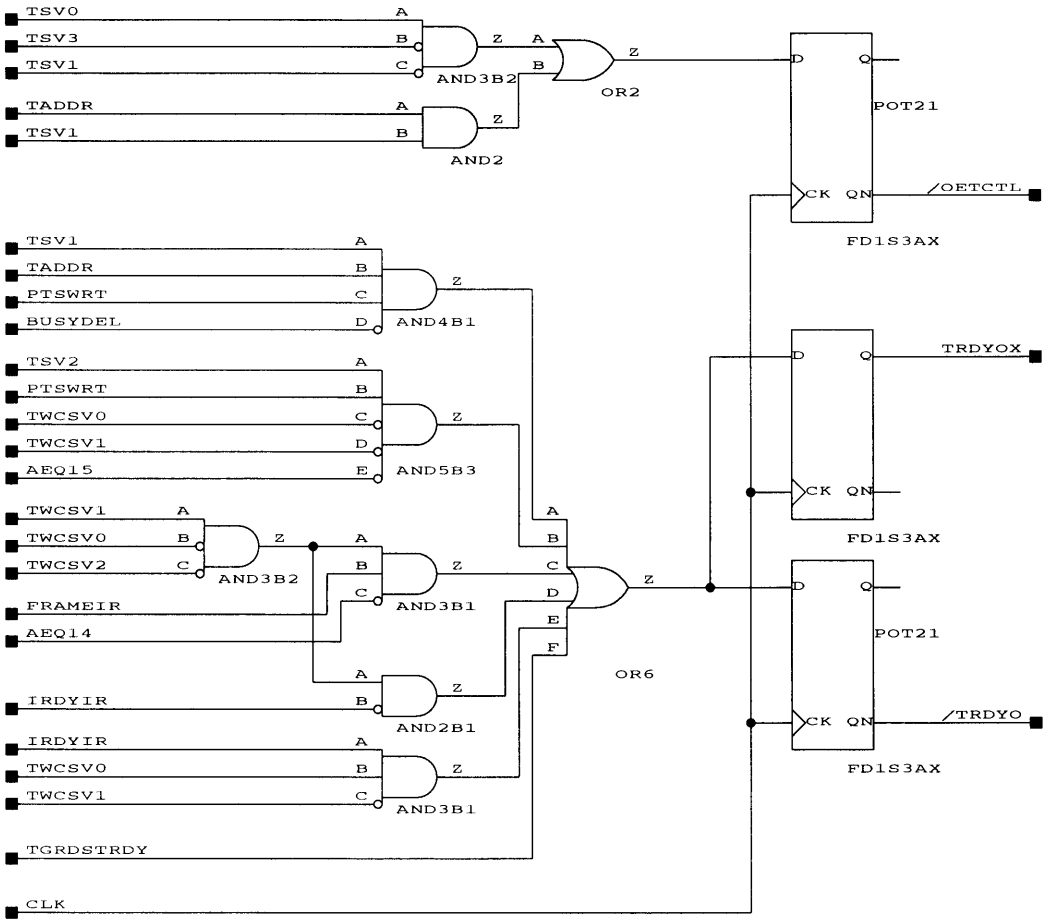


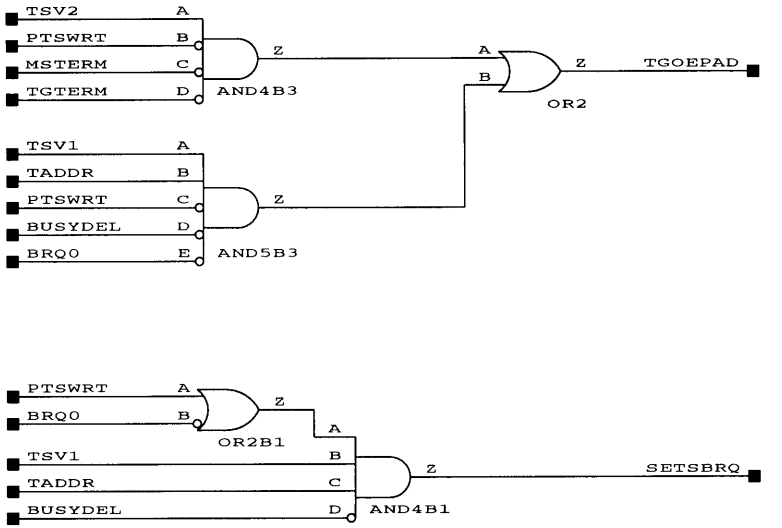

| | | |
|---|---|---|
| dde Dansk Data Elektronik A/S | | |
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI MASTER/TARGET CONTROL |
| Issue 2 | | Target Control Logic |
| Issue 3 | | File: pcimtctr Page:5 of 11 |



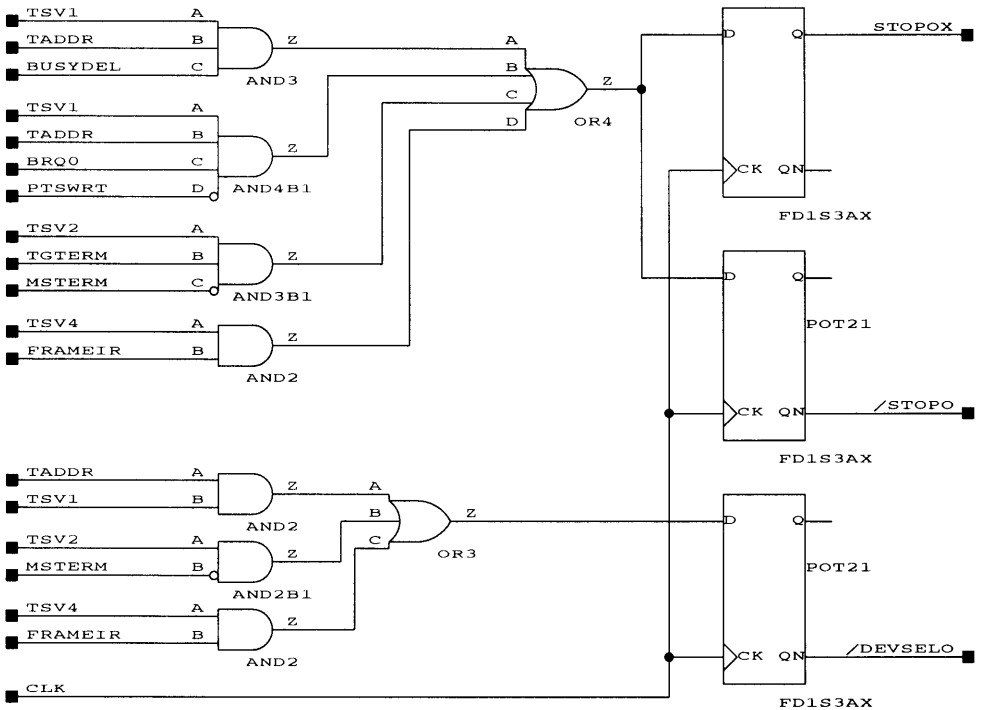




| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI MASTER/TARGET CONTROL |
| Issue 2 | | Target Control Logic |
| Issue 3 | | File: pcimtctr Page:6 of 11 |

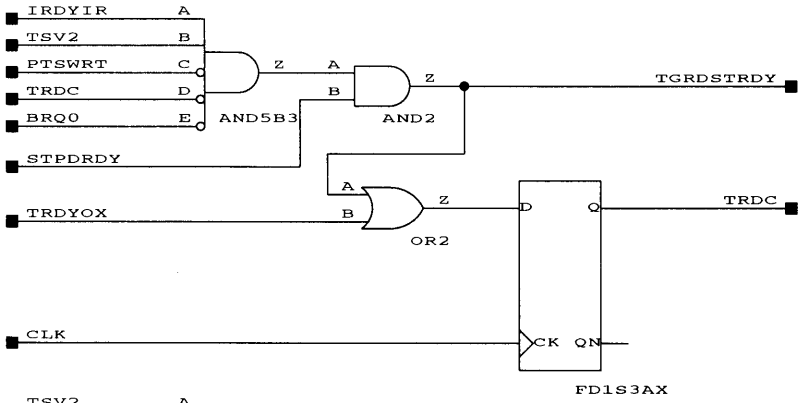


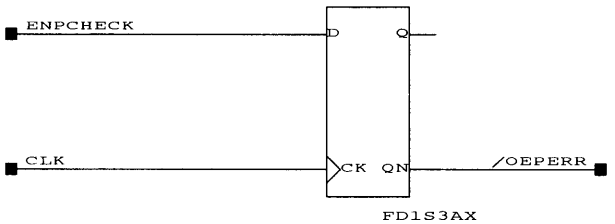



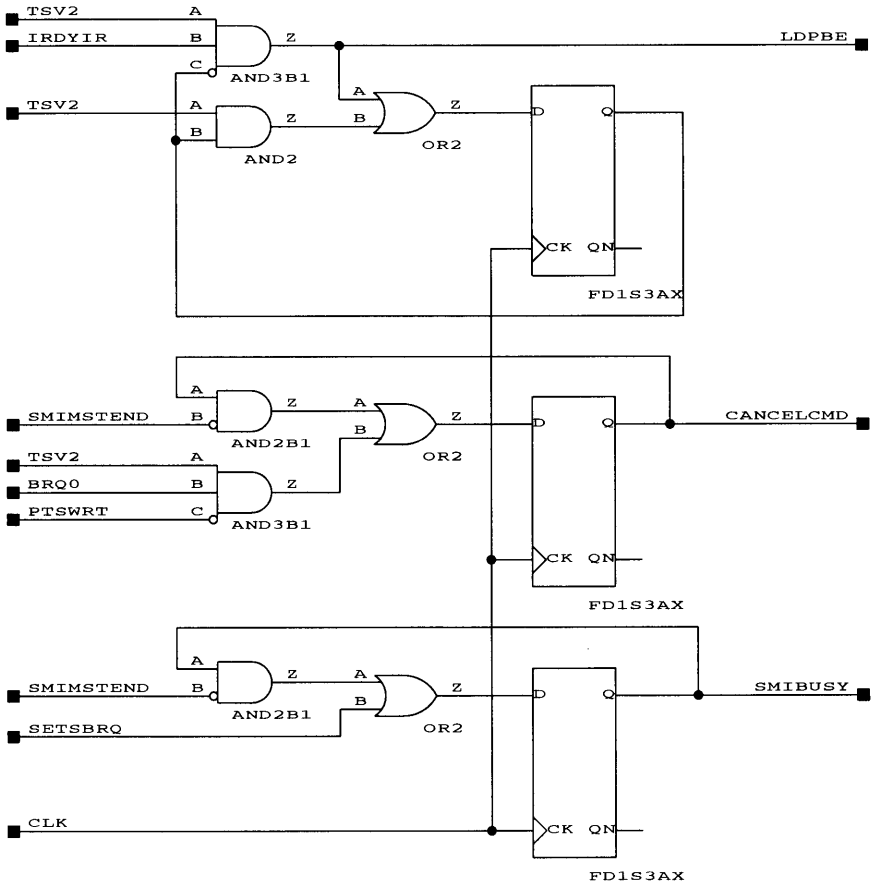

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI MASTER/TARGET CONTROL |
| Issue 2 | | Target Control Logic |
| Issue 3 | | File: pcimtctr Page:7 of 11 |



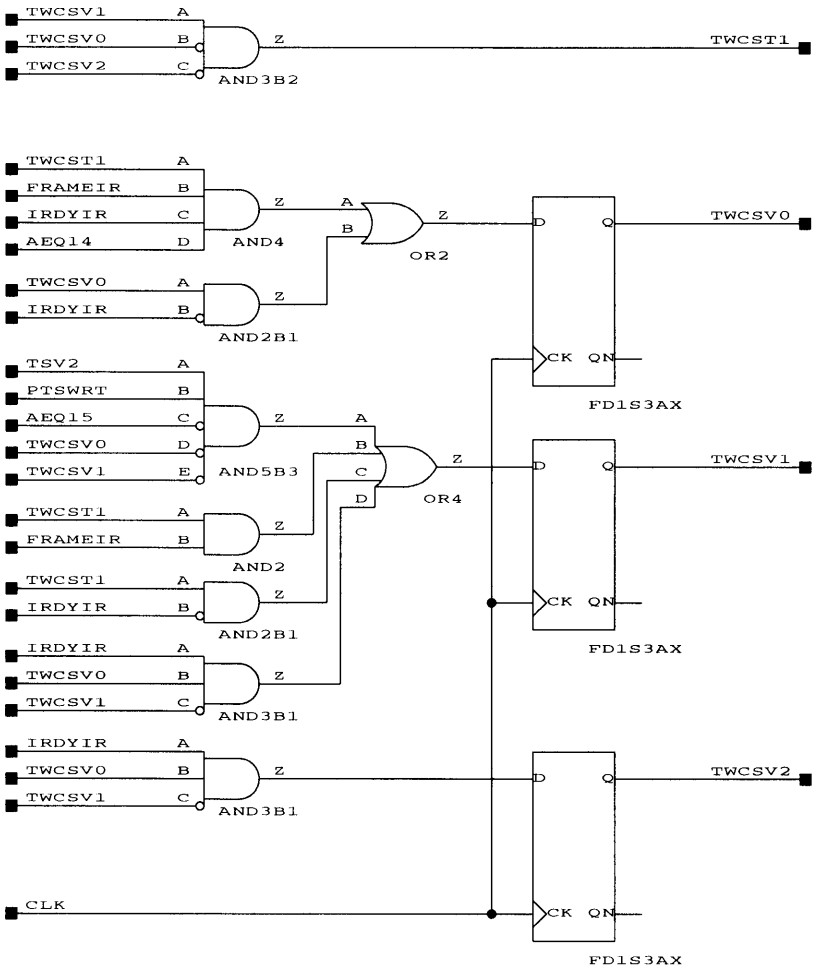

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970901 | BSP301-1 | |
| Issue 1 | | PCI MASTER/TARGET CONTROL | |
| Issue 2 | | Target Control Logic | |
| Issue 3 | | File: pcimtctr | Page:8 of 11 |

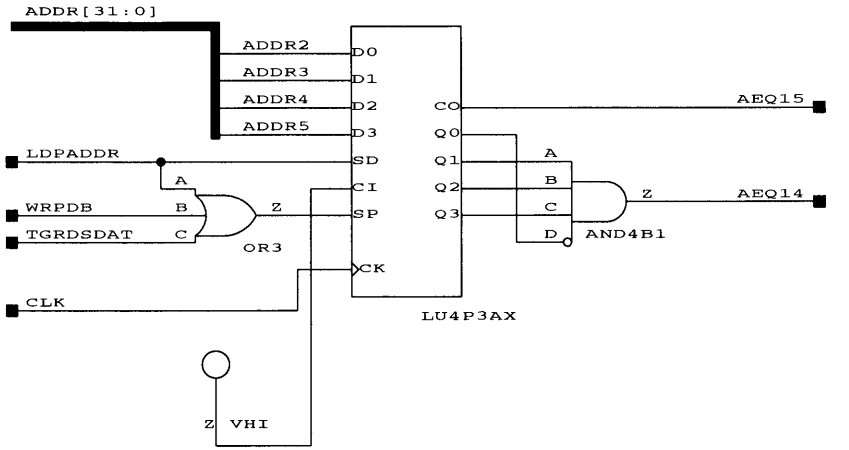


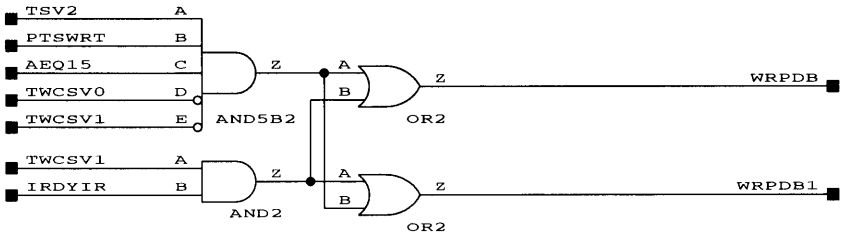



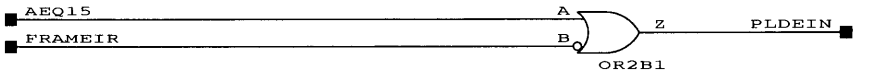




| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970917 | BSP301-1 |
| Issue 1 | | PCI MASTER/TARGET CONTROL |
| Issue 2 | | Target Control Logic |
| Issue 3 | | File: pcimtctr Page:9 of 11 |

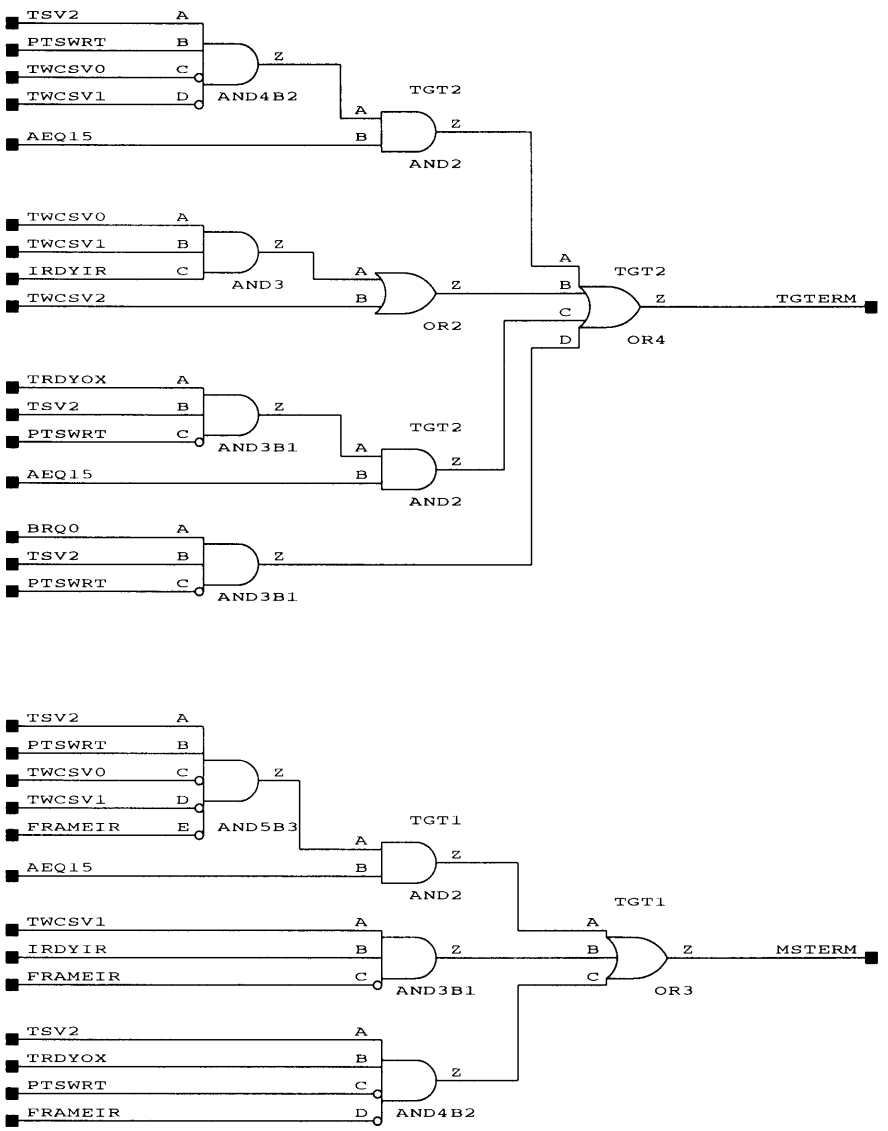


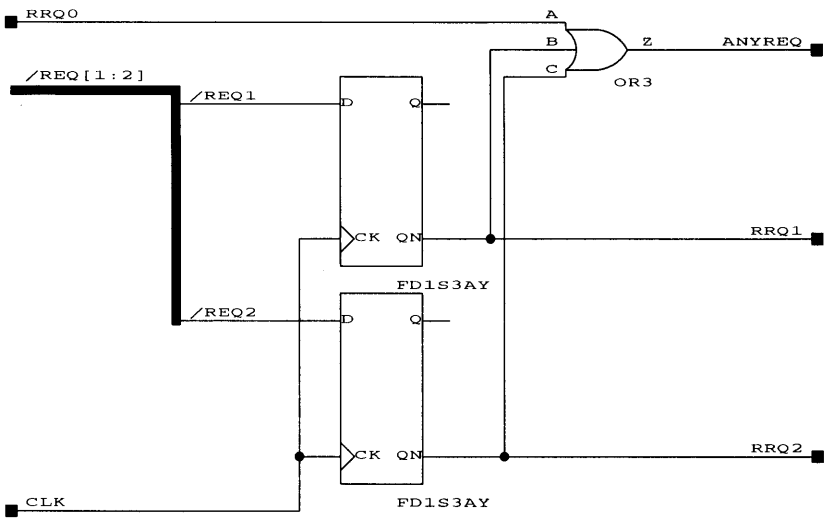



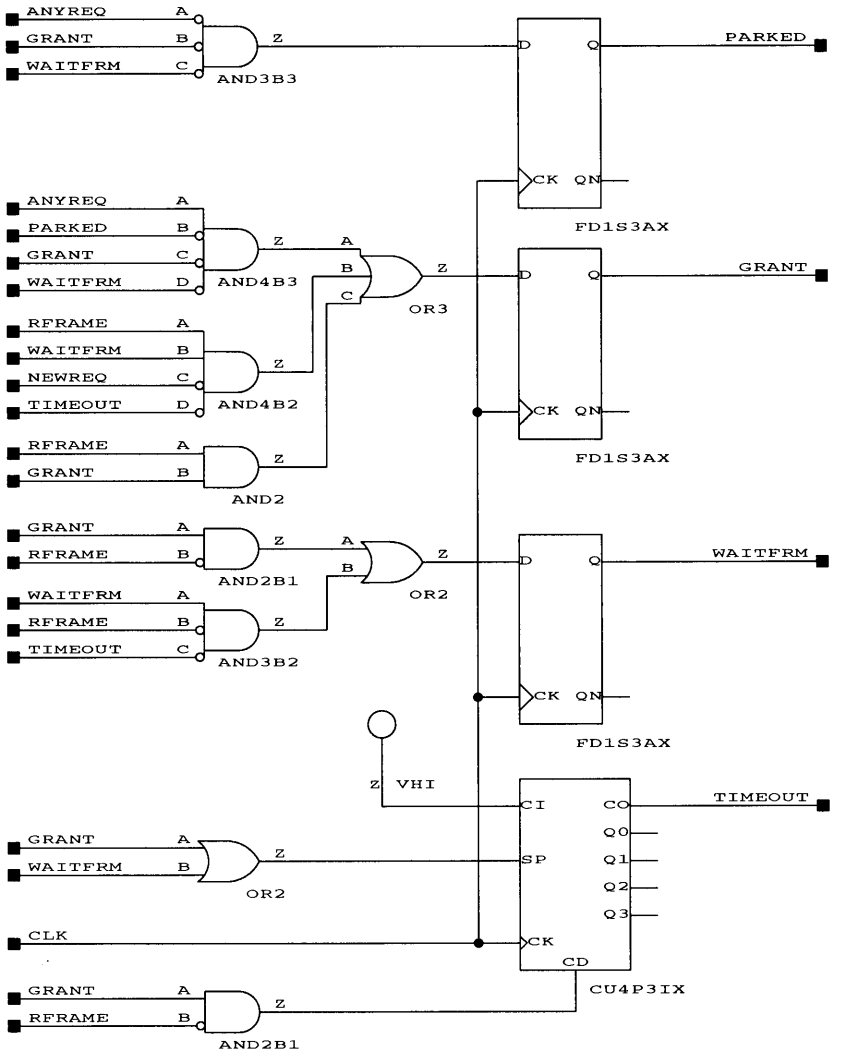




| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI MASTER/TARGET CONTROL |
| Issue 2 | | Target Control Logic |
| Issue 3 | | File: pcimtctr Page:10 of 11 |

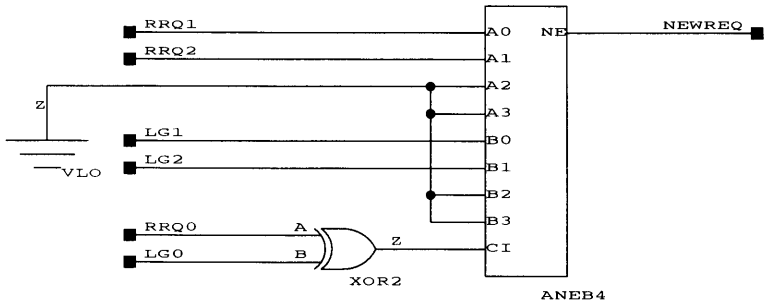


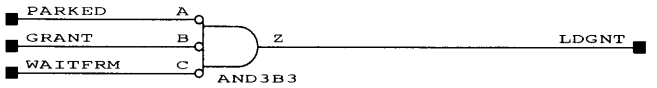

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 970901 | BSP301-1 |
| Issue 1 | | PCI MASTER/TARGET CONTROL |
| Issue 2 | | Target Control Logic |
| Issue 3 | | File: pcimtctr Page:11 of 11 |

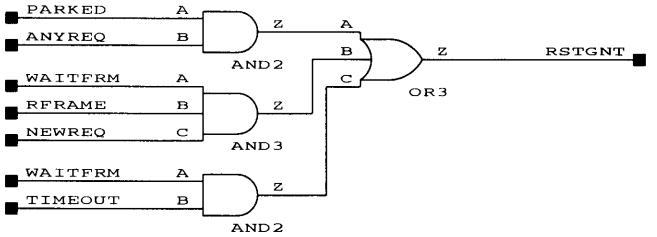


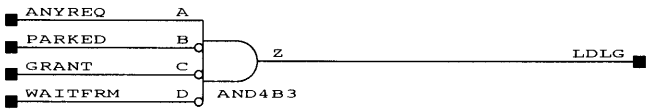




| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970917 | BSP301-1 PCI ARBITRATION CONTROL | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: pciarb | Page:1 of 4 |

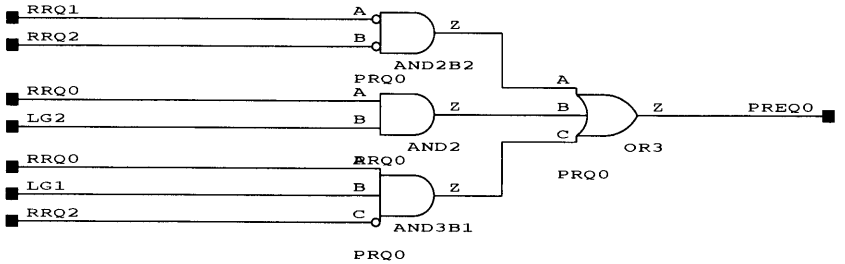


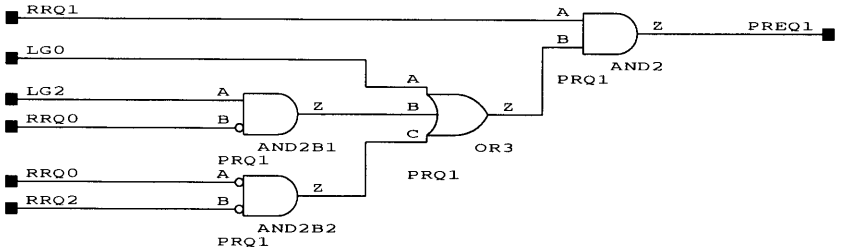



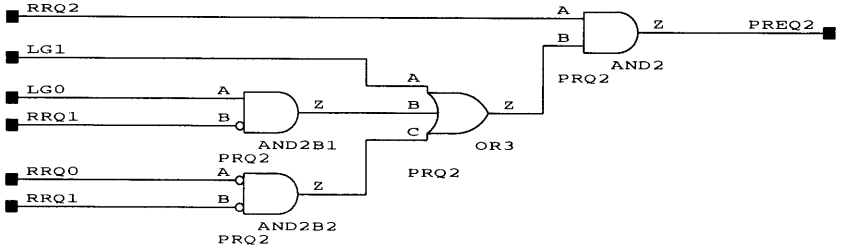



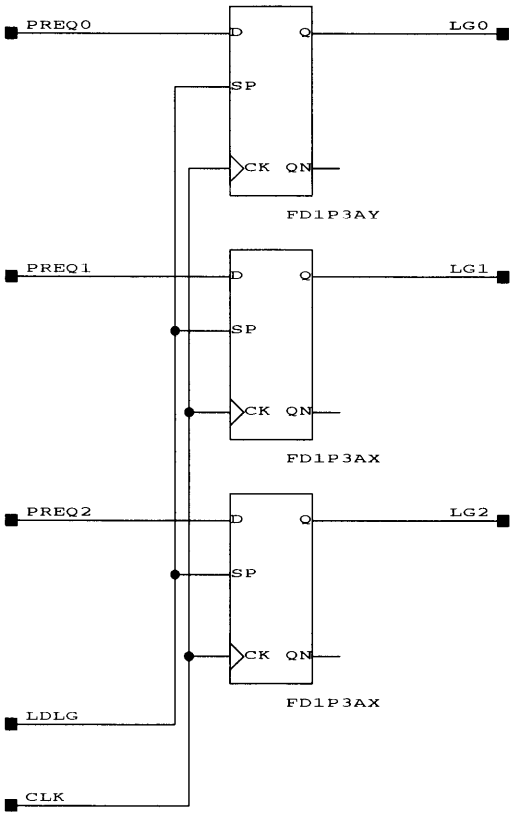

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970917 | BSP301-1<br>PCI ARBITRATION CONTROL | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: pciarb | Page:2 of 4 |

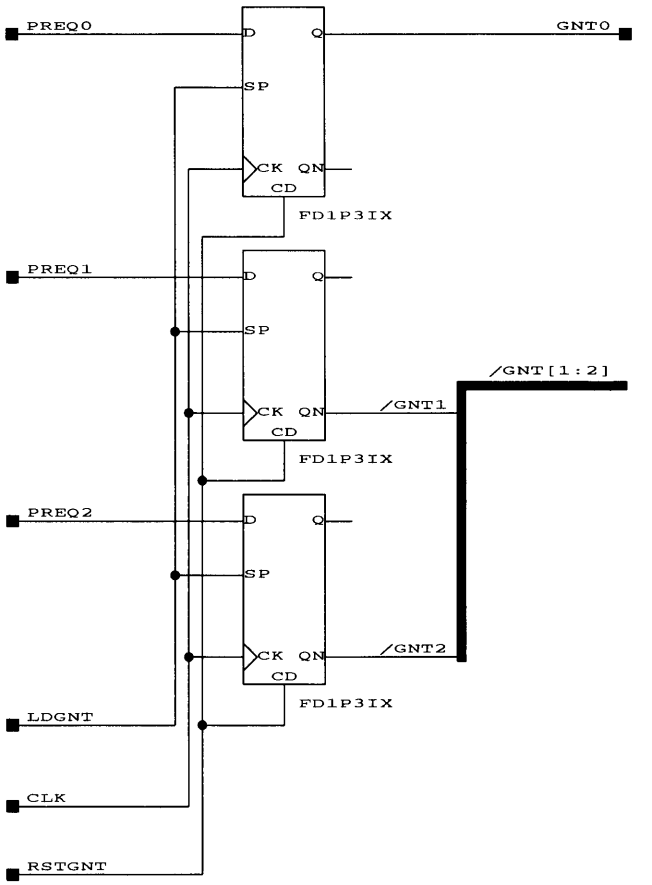



| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970917 | BSP301-1 PCI ARBITRATION CONTROL | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: pciarb | Page:3 of 4 |

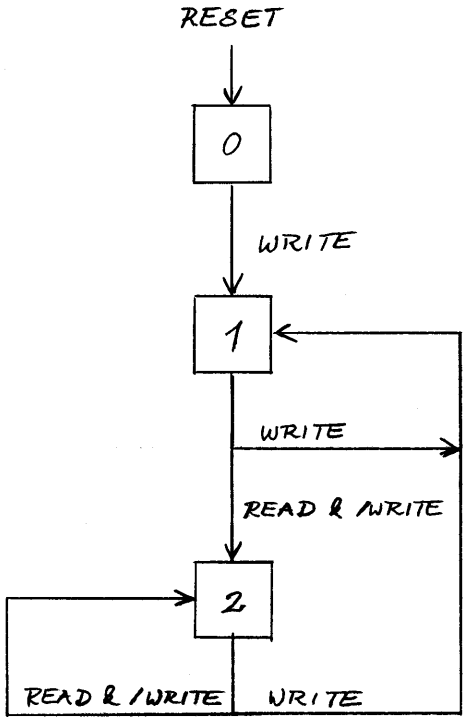



| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 970917 | BSP301-1 PCI ARBITRATION CONTROL | |
| Issue 1 | | | |
| Issue 2 | | | |
| Issue 3 | | File: pciarb | Page:4 of 4 |

AAJ / 1997-03-20

# DATA BUFFER CONTROL




LASTWR, LASTRD

0 0

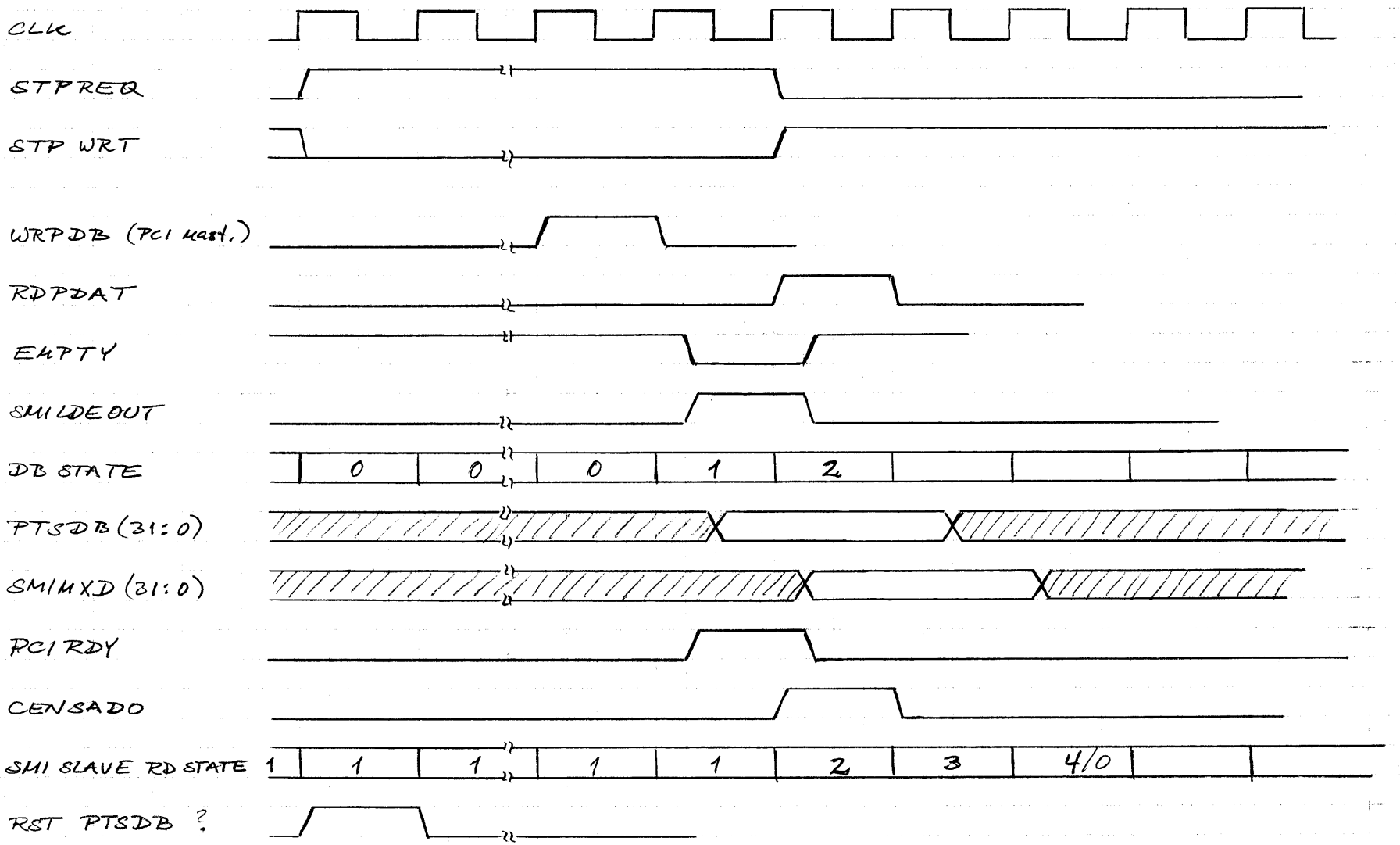
1 0

0 1

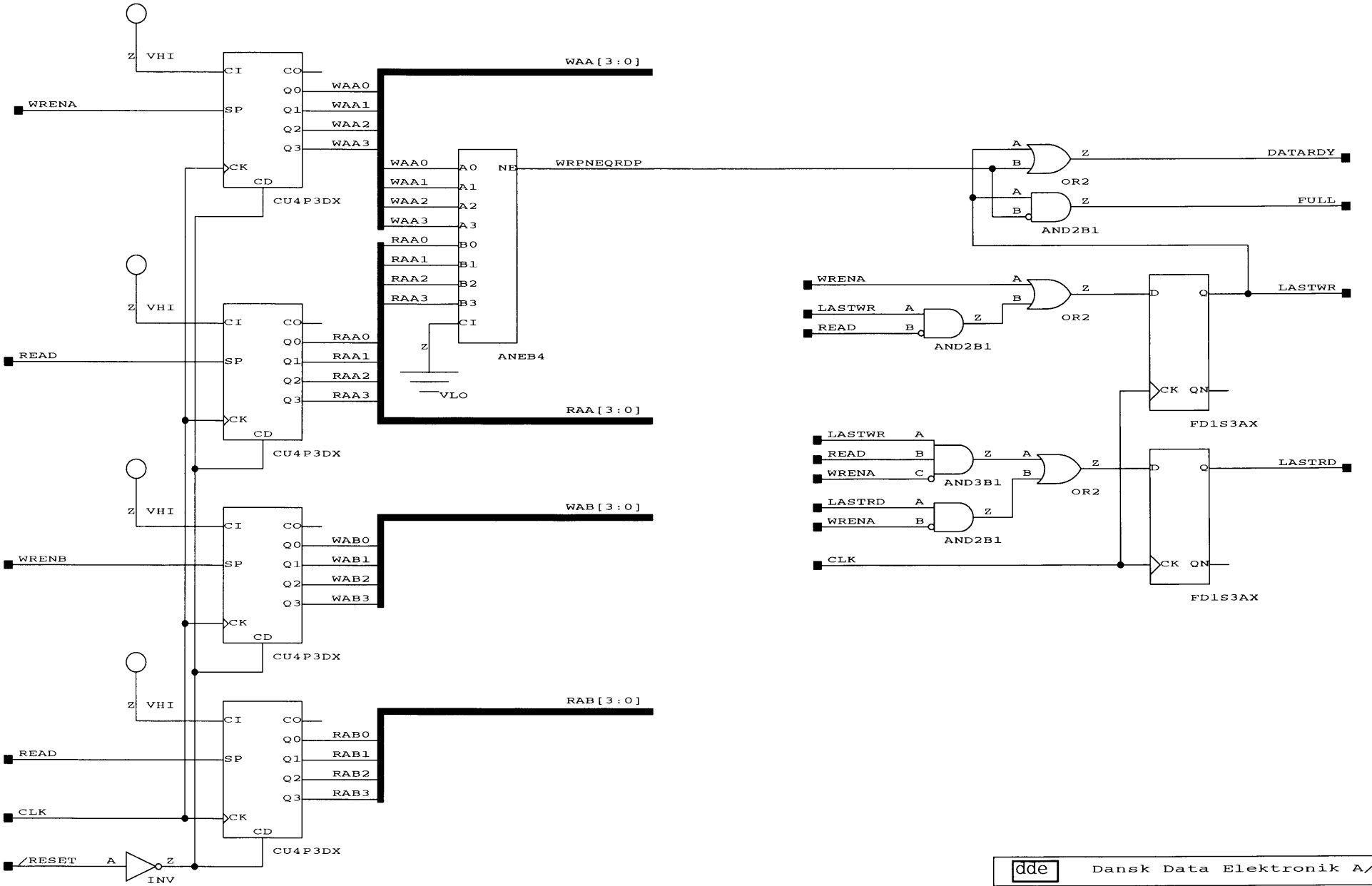
```
LASTWR := STATE0 & WRITE
        + STATE1 & /READ
        + STATE1 & WRITE
        + STATE2 & WRITE

LASTWR := WRITE
        + LASTWR & /READ

LASTRD := STATE1 & READ & /WRITE
        + STATE2 & /WRITE

LASTRD := LASTWR & READ & /WRITE
        + LASTRD & /WRITE

EMPTY = STATE0
      + STATE2 & (RDP = WRP)

EMPTY = /LASTWR & (RDP = WRP)

FULL  = LASTW & (RDP = WRP)
```

AAJ / 1997-03-21

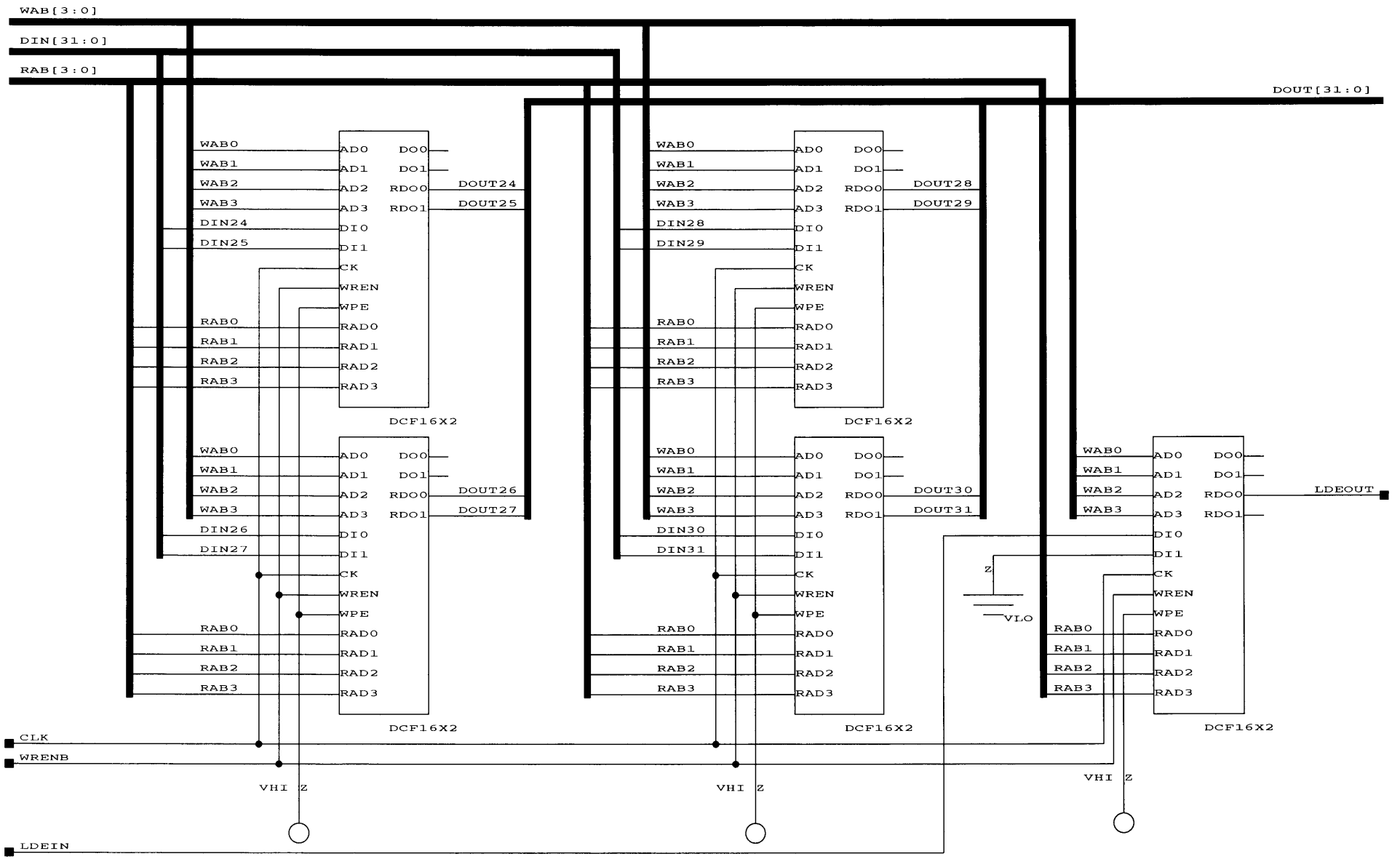
## DATA BUFFER CONTROL - READ FROM PCI TO SMI




NO PCI DEVSEL or PCI parity error causes SMI parity error.




| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 96-11-14 | BSP301-0 |
| Issue 1 | | DATA BUFFER - 16 WORDS |
| Issue 2 | | |
| Issue 3 | | File: databuf Page:5 of 5 |




| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 96-11-14 | BSP301-0 |
| Issue 1 | | DATA BUFFER - 16 WORDS |
| Issue 2 | | |
| Issue 3 | | File: databuf Page:4 of 5 |

# SMI DATA MULTIPLEXER CONTROL

ADDR(11:0)

| 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | RG0 + /RDID ; | ID-PROM |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | RG0 + /RDFCN ; | FCN-PROM |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | RG1 , SMI STATUS | |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | RG2 , SMI CONTROL | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RG3 , PCI STATUS | |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RG4 , PCI CONTROL | |
| ~~0~~ | ~~0~~ | ~~1~~ | ~~1~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~RG6 , PCI WINDOW BASE 1~~ | |
| ~~0~~ | ~~0~~ | ~~1~~ | ~~1~~ | ~~1~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~RG7 , PCI — — 2~~ | |
| ~~0~~ | ~~1~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~0~~ | ~~RG5 , PCI MASTER LATENCY~~ | |

RDID = /A7 & /A8 & /A9 & /A10

RDFCN = A7 & /A8 & /A9 & /A10

RG0 = RDID + RDFCN

| /GA[1:0] | | ADDR(31:24) | | |
|---|---|---|---|---|
| | | | 27 26 25 | |
| 10 | 10 | 0000 | 100 | POSITION 1 |
| 01 | 01 | 0000 | 101 | 2 |
| 00 | 00 | 0000 | 110 | 3 |

AAJ / 1997-08-08

# SMI DATA MULTIPLEXER CONTROL

| ADDR 9 8 7 | SEL 4 2 1 0 | EN BTW | |
|---|---|---|---|
| 0 0 X | 0 Φ 0 0 | 0 | RG0 |
| 0 1 0 | 0 Φ 0 1 | 0 | RG1 |
| 0 1 1 | 0 Φ 1 Φ | 0 | RG2 |
| 1 0 0 | 1 0 Φ Φ | 1 | RG3 |
| 1 0 1 | 1 1 Φ Φ | 0 | RG4 |

SEL0 = A8

SEL1 = A8 & A7

SEL2 = A7

SEL4 = A9

RDID = /A24 & /A23 & /A22 & /A21 & /A9 & /A8 & /A7

SMI ADDR

| 31 28 | 27 24 | |
|---|---|---|
| 0000 | 100X | POSITION 1 |
| 0000 | 101X | 2 |
| 0000 | 110X | 3 |

| 24 | 23 22 20 | |
|---|---|---|
| 0 | 00X | BSP301 internal registers |
| 0 | 010 | PCI INTR ACK |
| 0 | 011 | PCI CONFIG |
| 0 | 1XX | PCI I/O |
| 1 | XXX | PCI MEM |

BSP INTERNAL REGISTERS

| 24 | 23 21 | 10 9 8 | 7 | |
|---|---|---|---|---|
| 0 | 000 | 00 | 0 | RG0 + RD ID |
| | | 00 | 1 | RG0 + RD FCN |
| | | 01 | 0 | RG1, SMI STATUS |
| | | 01 | 1 | RG2, SMI CONTROL |
| | | 10 | 0 | RG3, PCI STATUS |
| | | 10 | 1 | RG4, PCI CONTROL |

# SMI DATA MULTIPLEXER CONTROL

| ADDR 10 9 8 7 | SEL 5 4 3 2 1 0 | EN 8T31 | EN 0T7 | |
|---|---|---|---|---|
| 0 0 0 / | / 0 / / 0 0 | 0 | 1 | RG0 |
| 0 0 1 0 | / 0 / / 0 1 | 0 | 1 | RG1 |
| 0 0 1 1 | / 0 / / 1 / | 0 | 1 | RG2 |
| 0 1 0 0 | / 1 0 0 / / | 0 | 1 | RG3 |
| 0 1 0 1 | / 1 0 1 / / | 0 | 1 | RG4 |
| ~~0 1 1 0~~ | ~~0 / / / / /~~ | ~~1~~ | ~~0~~ | ~~RG6~~ |
| ~~0 1 1 1~~ | ~~1 / / / / /~~ | ~~1~~ | ~~0~~ | ~~RG7~~ |
| ~~1 0 0 0~~ | ~~/ 1 1 / / /~~ | ~~0~~ | ~~1~~ | ~~RG5~~ |

SEL 5 = A7 (A9 & A7)

SEL 4 = A10 + A9

SEL 3 = A10 (A10 & /A8)

SEL 2 = A7 (A9 & A7)

SEL 1 = A8 & A7

SEL 0 = A8 (/A10 & A8)

EN8T31 = /A10 & A9 & A8

EN0T7 = /EN8T31

RDID = /A7 (/A10 & /A7)

RDFCN = A7 (/A10 & A7)

## SEL(0:5) DECODING FROM ADDRESS(10:7)

| 10,9 \ 8,7 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | Φ | Φ | Φ | Φ |
| 01 | Φ | Φ | 1 | 0 |
| 11 | Φ | Φ | Φ | Φ |
| 10 | Φ | Φ | Φ | Φ |

SEL5

| 10,9 \ 8,7 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 1 | 1 | Φ | Φ |
| 11 | Φ | Φ | Φ | Φ |
| 10 | 1 | Φ | Φ | Φ |

SEL4

| 10,9 \ 8,7 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | Φ | Φ | Φ | Φ |
| 01 | 0 | 0 | Φ | Φ |
| 11 | Φ | Φ | Φ | Φ |
| 10 | 1 | Φ | Φ | Φ |

SEL3

| 10,9 \ 8,7 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | Φ | Φ | Φ | Φ |
| 01 | 0 | 1 | Φ | Φ |
| 11 | Φ | Φ | Φ | Φ |
| 10 | Φ | Φ | Φ | Φ |

SEL2

| 10,9 \ 8,7 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 0 |
| 01 | Φ | Φ | Φ | Φ |
| 11 | Φ | Φ | Φ | Φ |
| 10 | Φ | Φ | Φ | Φ |

SEL1

| 10,9 \ 8,7 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | Φ | 1 |
| 01 | Φ | Φ | Φ | Φ |
| 11 | Φ | Φ | Φ | Φ |
| 10 | Φ | Φ | Φ | Φ |

SEL0

1997-02-07/AAJ
BSP301

## SMI-TO-PCI COMMAND ENCODING

| SMI ADDR (24:21) | PCI CMD(3:0) | COMMAND |
|---|---|---|
| 0010 | 0000 | Interrupt ACK |
| 0011 | 1010 | Configuration Read |
| | 1011 | — Write |
| 01XX | 0010 | I/O Read |
| | 0011 | I/O Write |
| 1XXX | 0110 | Memory Read |
| | 0111 | Memory Write |

PCI CMD(0) = WRITE

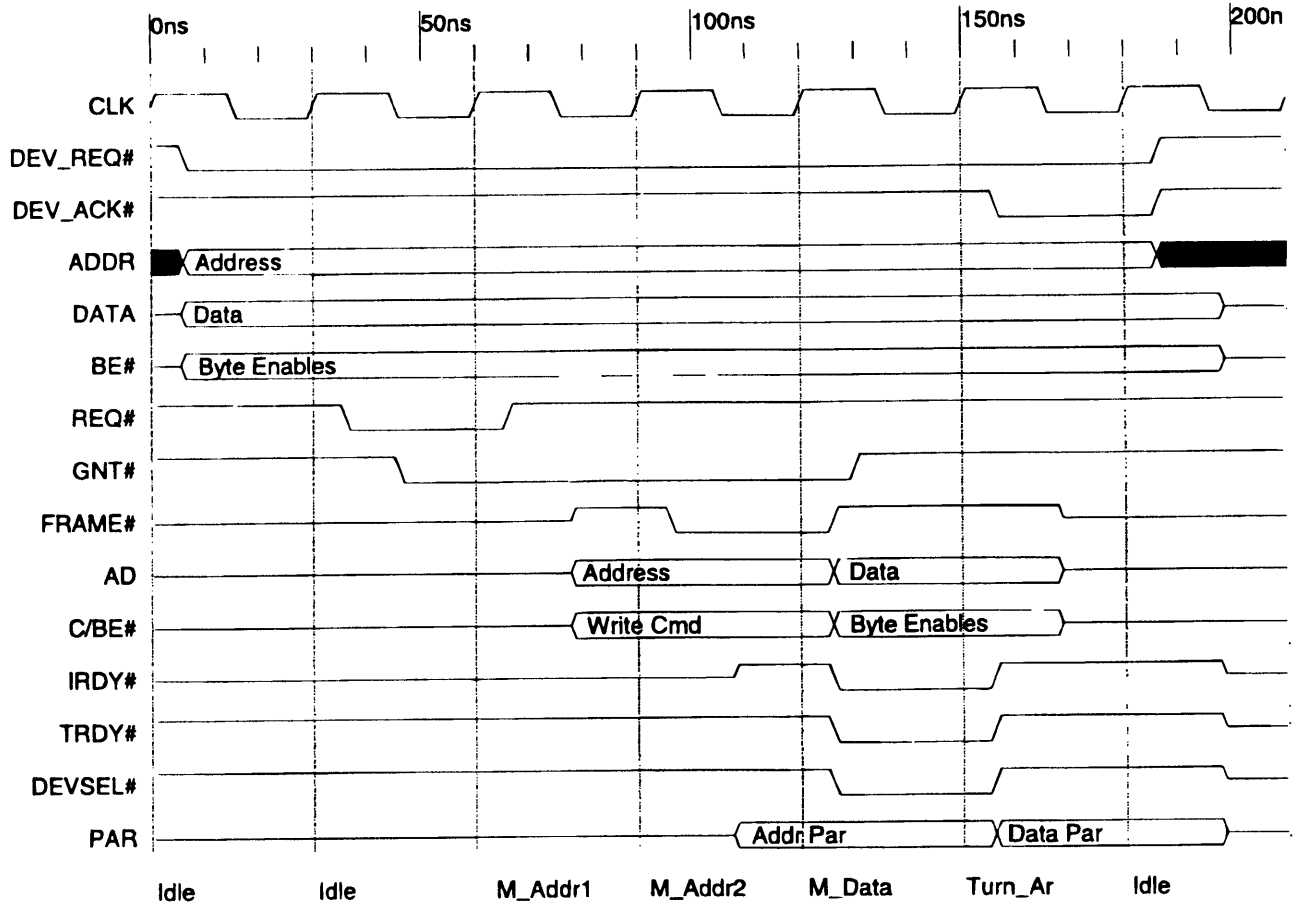
PCI CMD(1) = A24
+ A23
+ A22 & A21

PCI CMD(2) = A24

PCI CMD(3) = /A24 & /A23 & A22 & A21

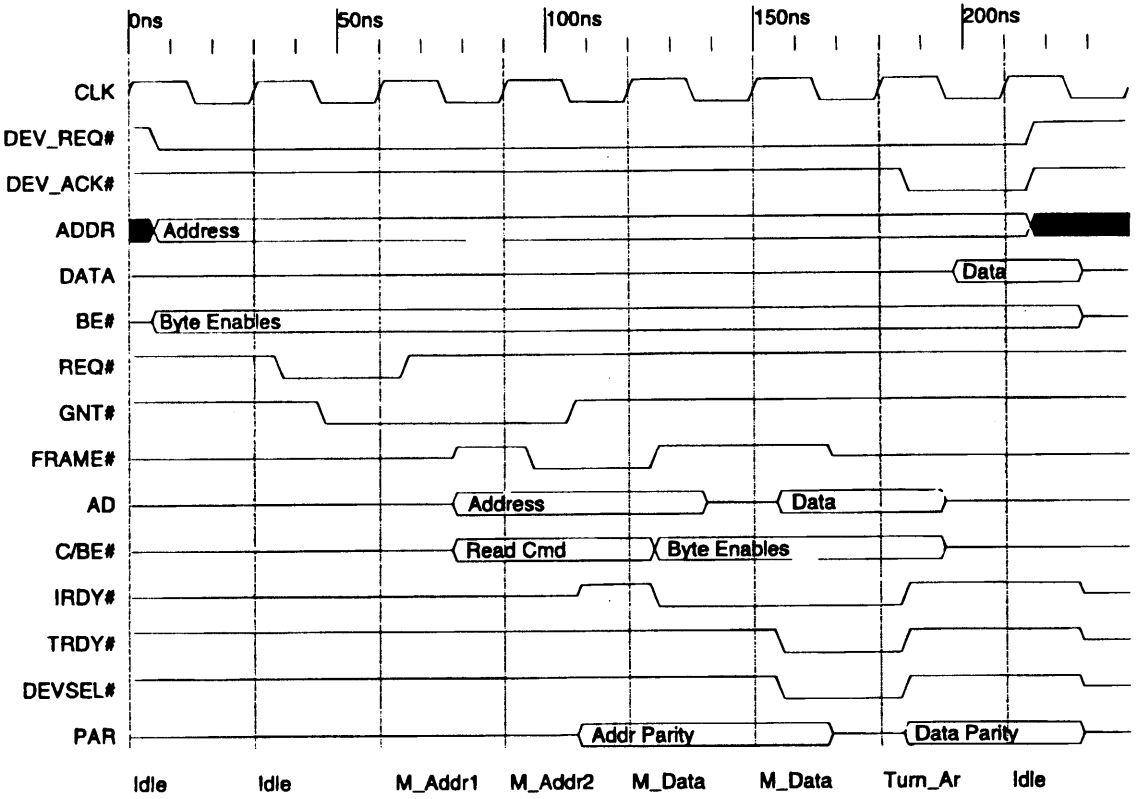

# The PCI Interface

## PCI Write Transaction




# The PCI Interface

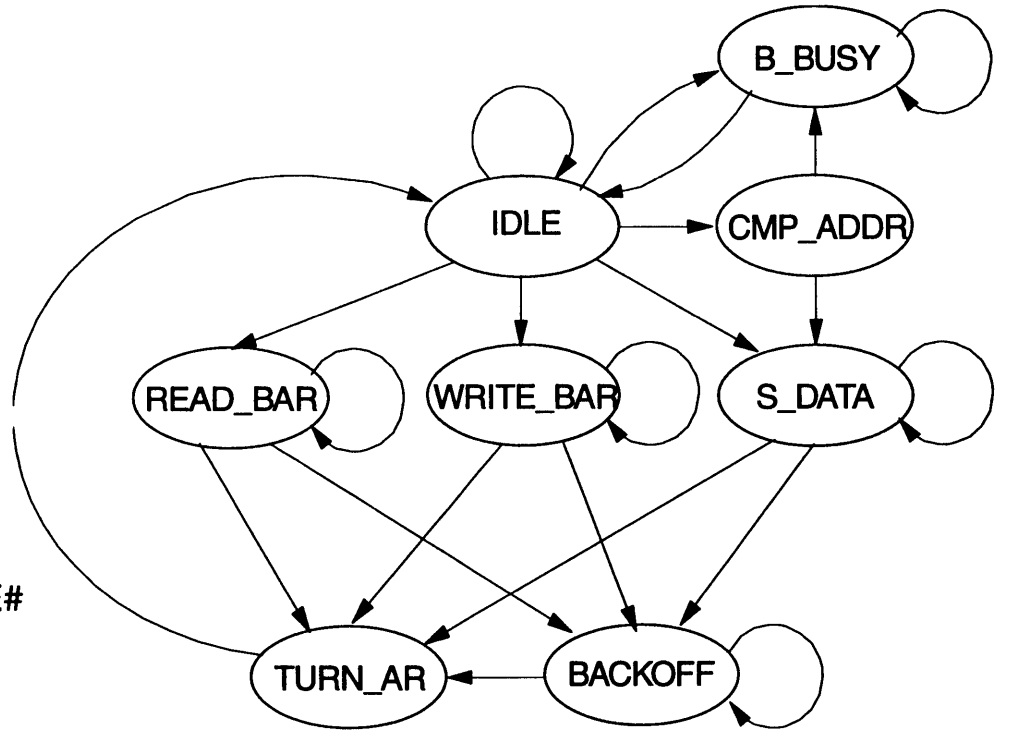
## PCI Read Transaction




# The PCI Interface

## The PCI Target Controller

- CMP_ADDR
  - Address Comparison
- READ_BAR (WRITE_BAR)
  - Configuration Accesses
- S_DATA
  - Target transfers data
- TURN_AR
  - Bus turn-around cycle
- BACKOFF
  - Waits for master to deassert FRAME#

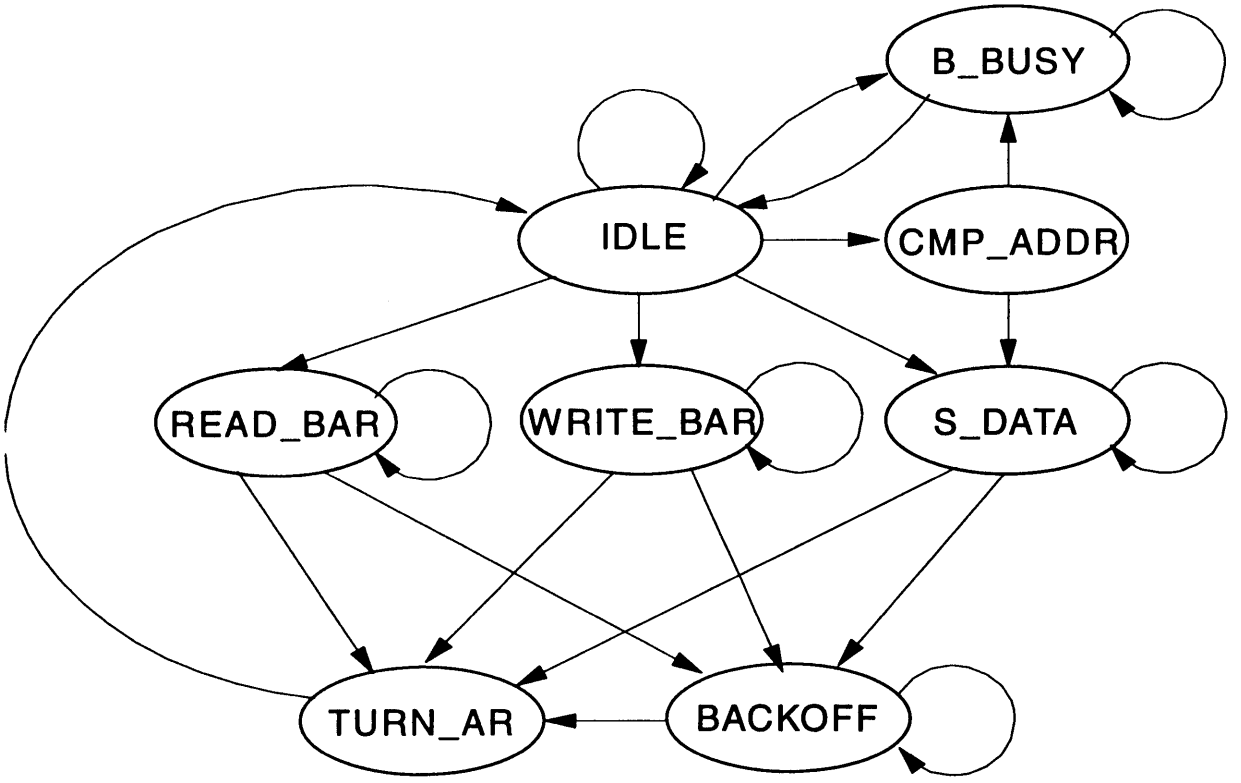


DESIGN·FOR·SUCCESS

# Sample Design

## Target State Diagram

- The target controller can be easy modified for custom applications



| | |
|---|---|
| REQ4N | 30 |
| RESETN | 106 |
| SAD0 | 143 |
| SAD1 | 142 |
| SAD10 | 131 |
| SAD11 | 129 |
| SAD12 | 128 |
| SAD13 | 127 |
| SAD14 | 126 |
| SAD15 | 124 |
| SAD16 | 120 |
| SAD17 | 119 |
| SAD18 | 118 |
| SAD19 | 115 |
| SAD2 | 141 |
| SAD20 | 114 |
| SAD21 | 113 |
| SAD22 | 111 |
| SAD23 | 110 |
| SAD24 | 100 |
| SAD25 | 99 |
| SAD26 | 97 |
| SAD27 | 96 |
| SAD28 | 95 |
| SAD29 | 92 |
| SAD3 | 140 |
| SAD30 | 91 |
| SAD31 | 90 |
| SAD4 | 139 |
| SAD5 | 138 |
| SAD6 | 137 |
| SAD7 | 136 |
| SAD8 | 133 |
| SAD9 | 132 |
| SERRN | 190 |
| STOPN | 192 |
| TRDYN | 194 |
| WR | 68 |
| WRN | 69 |

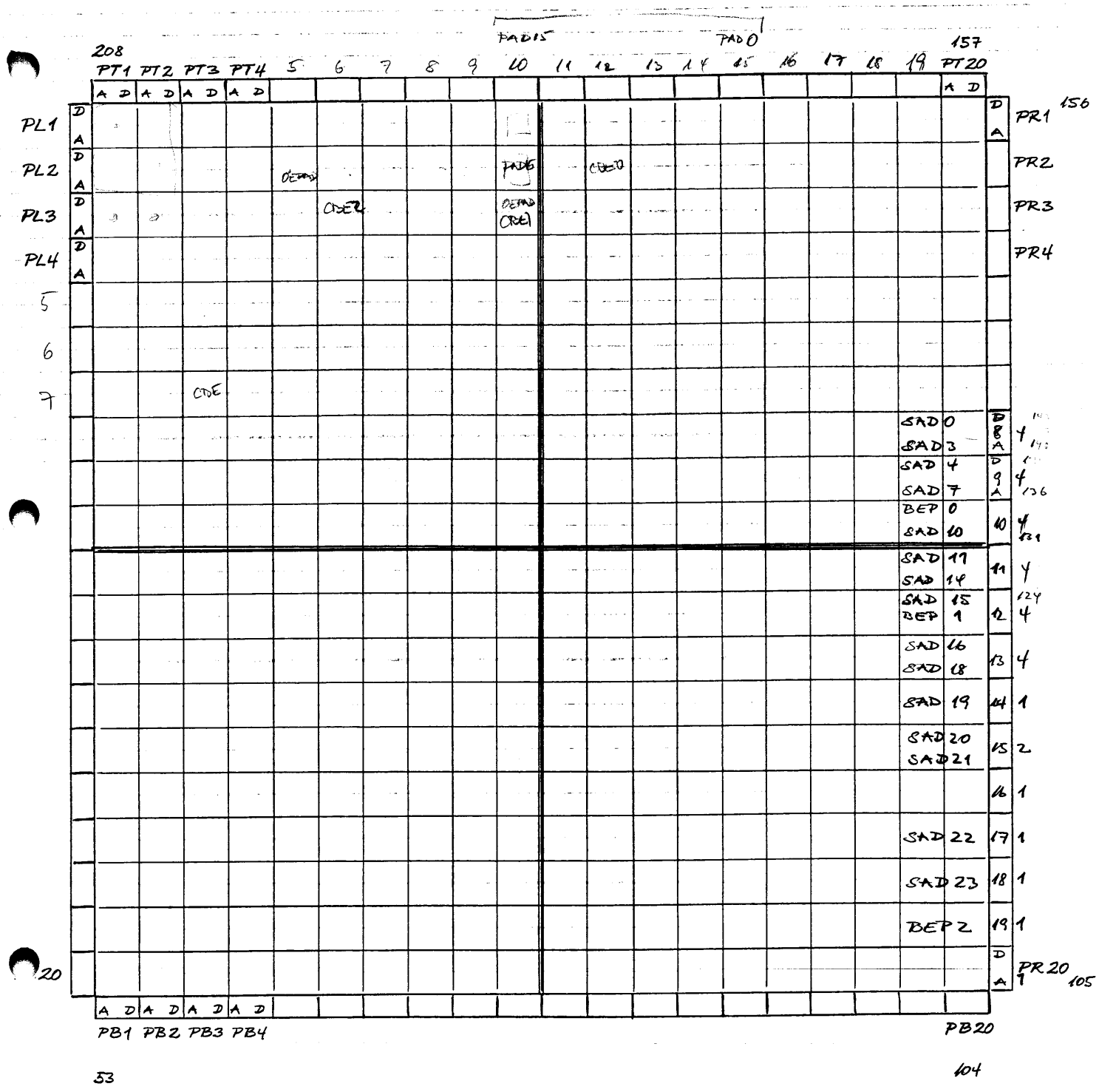
Pinout by Pin Number:

| Pin Number | | Comp Name | | Preference | | POS |
|---|---|---|---|---|---|---|
| 3 | PL 1 | | | | | |
| 4 | PL 2 | | | | | |
| 5 | | PAD20 | | LOCATED | | |
| 6 | PL 4 | PAD21 | 11 | LOCATED | √ √ | R4C1 / C2 |
| 7 | PL 5 | PAD22 | 12 | LOCATED | √ √ | R5C1 / C2 |
| 8 | PL 6 | PAD23 | 13 | LOCATED | √ | R6C1 / C2 |
| 9 | | CBE3 | | LOCATED | √ | |
| 10 | PL 7 | PAD24 | 14 | LOCATED | √ √ | R7C1 / C2 |
| 11 | | PAD25 | | LOCATED | | |
| 13 | | PAD26 | | LOCATED | √ | |
| 14 | PL 8 | PAD27 | 15 | LOCATED | √ | R8C1 / C2 |

| | | | | | |
|---|---|---|---|---|---|
| 15 | PL 8 | PAD28 | 15 | LOCATED JJ | R8C1 /C2 |
| 16 | | PAD29 | | LOCATED | |
| 17 | PL 9 | PAD30 | 16 | LOCATED | R9C1 /C2 |
| 18 | | PAD31 | | LOCATED J | |
| 19 | | INTAN | | LOCATED | |
| 20 | | INTBN | | LOCATED | |
| 22 | PL 10 | CLK | | LOCATED | R10C1 /C2 |
| 23 | | | | | |
| 24 | | INTCN | | LOCATED | |
| 25 | | INTDN | | LOCATED | |
| 27 | PL 11 | REQ1N | | LOCATED | R11C1 /C2 |
| 28 | | REQ2N | | LOCATED | |
| 29 | | REQ3N | | LOCATED | |
| 30 | | REQ4N | | LOCATED | |
| 32 | PL 12 | GNTN1 | | LOCATED | R12C1 /C2 |
| 33 | | GNTN2 | | LOCATED | |
| 34 | | GNTN3 | | LOCATED | |
| 35 | | GNTN4 | | LOCATED | |
| 36 | PL 13 | | | | |
| 37 | | | | | |
| 38 | | | | | |
| 39 | | | | | |
| 41 | PL 14 | | | | |
| 42 | | | | | |
| 43 | PL 15 | | | | |
| 44 | | | | | |
| 45 | PL 16 | | | | |
| 46 | PL 17 | | | | |
| 47 | PL 18 | | | | |
| 48 | PL 19 | | | | |
| 49 | | | | | |
| 50 | PL 20 | | | | |
| 55 | PB 1 | | | | |
| 56 | PB 2 | | | | |
| 57 | | | | | |
| 58 | PB 3 | | | | |
| 59 | PB 4 | | | | |
| 60 | PB 5 | GA1N | | LOCATED | R20C5/ R19 |
| 61 | PB 6 | GA0N | | LOCATED | R20C6/ |
| 62 | | IRQ | S20 | LOCATED | |
| 63 | PB 7 | BGOUTN J | | LOCATED | R20/C7 |
| 64 | | BGINN | S21 | LOCATED | |
| 66 | PB 8 | BUSY J | | LOCATED | R20C8 |
| 67 | | BRQ | | LOCATED | |
| 68 | | WR J | S22 | LOCATED | |
| 69 | | WRN | | LOCATED | |
| 70 | PB 9 | ACK J | | LOCATED | R20C9 |
| 71 | | ACKN | | LOCATED | |
| 72 | | DS J | S23 | LOCATED | |
| 73 | | DSN | | LOCATED | |
| 75 | PB 10 | ASACK J | | LOCATED- J | R20C10 |
| 76 | | ASACKN J | | LOCATED | |
| 77 | | ACN | S24 | LOCATED | |
| 78 | | AC J | | LOCATED | |
| 80 | PB 11 | PAS J | | LOCATED | R20C11 |
| 81 | | PASN | S25 | LOCATED | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 82 | PB 11 | ENEDRVN | S25 | LOCATED | | R20 C11 |
| 83 | | ENEREC | | LOCATED | | |
| 85 | | ERRLEDN | | LOCATED | | |
| 86 | PB 12 | | | | | R20 C12 |
| 87 | | | | | | |
| 88 | | BEP3 | 16 | LOCATED | J J | |
| 89 | | | | | | R20 C13 / R19 C13 |
| 90 | PB 13 | SAD31 | | LOCATED | | |
| 91 | | SAD30 | 15 | LOCATED | J J | |
| 92 | | SAD29 | | LOCATED | | |
| 94 | PB 14 | | | | | R20 C14 |
| 95 | | SAD28 | 14 | LOCATED | J J | |
| 96 | PB 15 | SAD27 | 13 | LOCATED | J J | R20 C15 |
| 97 | | SAD26 | | LOCATED | | |
| 98 | PB 16 | | | | | |
| 99 | PB 17 | SAD25 | 12 | LOCATED | J J | R20 C17 |
| 100 | PB 18 | SAD24 | 11 | LOCATED | J J | R20 C18 / R19 |
| 101 | | | | | | |
| 102 | | | | | | |
| 106 | | RESETN | | | | |
| 108 | | | | | | |
| 109 | PR 19 | BEP2 | 10 | LOCATED | J | R19 C20 / C19 |
| 110 | PR 18 | SAD23 | 9 | LOCATED | J | R18 C20 |
| 111 | PR 17 | SAD22 | 8 | LOCATED | J J | R17 C20 |
| 112 | PR 16 | | | | | |
| 113 | PR 15 | SAD21 | 7 | LOCATED | J J | R15 C20 |
| 114 | | SAD20 | | LOCATED | | |
| 115 | PR 14 | SAD19 | 6 | LOCATED | J J | R14 C20 |
| 117 | | | | | | |
| 118 | PR 13 | SAD18 | 5 | LOCATED | J J | R13 C20 / C19 |
| 119 | | SAD17 | | LOCATED | | |
| 120 | | SAD16 | | LOCATED | | |
| 121 | | | | | | |
| 122 | PR 12 | | | | | |
| 123 | | BEP1 | 4 | LOCATED | J J | R12 C20 |
| 124 | | SAD15 | | LOCATED | | |
| 126 | | SAD14 | | LOCATED | | |
| 127 | PR 11 | SAD13 | 3 | LOCATED | J J | R11 C20 |
| 128 | | SAD12 | | LOCATED | | |
| 129 | | SAD11 | | LOCATED | | |
| 131 | | SAD10 | | LOCATED | | |
| 132 | PR 10 | SAD9 | 2 | LOCATED | J J | R10 C20 |
| 133 | | SAD8 | | LOCATED | | |
| 134 | | BEP0 | | LOCATED | | |
| 136 | | SAD7 | | LOCATED | | |
| 137 | PR 9 | SAD6 | 1 | LOCATED | J J | R9 C20 |
| 138 | | SAD5 | | LOCATED | | |
| 139 | | SAD4 | | LOCATED | | |
| 140 | | SAD3 | | LOCATED | | |
| 141 | PR 8 | SAD2 | 0 | LOCATED | J J | R8 C20 / C19 |
| 142 | | SAD1 | | LOCATED | | |
| 143 | | SAD0 | | LOCATED | | |
| 145 | PR 7 | | | | | R7 C20 |
| 146 | | AUXD7 | | LOCATED | | |
| 147 | PR 6 | AUXD6 | | LOCATED | | R6 C20 |
| 148 | | AUXD5 | | LOCATED | | |

| | | | | | |
|---|---|---|---|---|---|
| 149 | PR5 | AUXD4 | | LOCATED | R5 C20 |
| 150 | | AUXD3 | | LOCATED | |
| 151 | PR4 | AUXD2 | | LOCATED | R4 C20 |
| 152 | PR3 | AUXD1 | | LOCATED | R3 C20 |
| 153 | PR2 | AUXD0 | | LOCATED | R2 C20 |
| 154 | PR1 | OEFCNN | | LOCATED | R1 C20 |
| 159 | PT20 | OEIDN | | LOCATED | R1 C20 |
| 160 | PT19 | AUXA4 | | LOCATED | R1 C19 |
| 161 | PT17 | AUXA3 | | LOCATED | R1 C17 |
| 162 | | AUXA2 | | LOCATED | |
| 163 | PT16 | AUXA1 | | LOCATED | R1 C16 |
| 164 | PT15 | AUXA0 | | LOCATED | |
| 165 | | PAD0 | 0 | LOCATED | R1 C15 / R2 C15 |
| 166 | PT14 | PAD1 | | LOCATED | R1 C14 |
| 167 | | PAD2 | 1 | LOCATED | |
| 169 | | PAD3 | | LOCATED | |
| 170 | PT13 | PAD4 | | LOCATED | R1 C13 |
| 171 | | PAD5 | 2 | LOCATED | |
| 172 | | PAD6 | | LOCATED | |
| 173 | | PAD7 | | LOCATED | |
| 174 | PT12 | CBE0 | | LOCATED | R1 C12 / R2 C12 |
| 175 | | PAD8 | 3 | LOCATED | |
| 176 | | PAD9 | | LOCATED | |
| 178 | | PAD10 | | LOCATED | |
| 179 | PT11 | PAD11 | | LOCATED | R1 C11 |
| 180 | | PAD12 | 4 | LOCATED | |
| 181 | | PAD13 | | LOCATED | |
| 183 | | PAD14 | | LOCATED | |
| 184 | PT10 | PAD15 | | LOCATED | R1 C10 / R2 C10 |
| 185 | | CBE1 | 5 | LOCATED | |
| 186 | | OEPAD1 | | | |
| 188 | | PAR | | LOCATED | |
| 189 | PT9 | PERRN | 20 | LOCATED | R1 C9 |
| 190 | | SERRN | | LOCATED | |
| 191 | | | | | |
| 192 | | STOPN | | LOCATED | |
| 193 | PT8 | DEVSELN | 21 | LOCATED | R1 C8 |
| 194 | | TRDYN | | LOCATED | |
| 195 | | | | | |
| 197 | PT7 | IRDYN | 22 | LOCATED | R1 C7 |
| 198 | | FRAMEN | | LOCATED | |
| 199 | | CBE2 | | LOCATED | |
| 200 | PT6 | OEPAD2 | 6 | | R1 C6 / R2 C6 |
| 201 | PT5 | PAD16 | 7 | LOCATED | R1 C5 |
| 202 | PT4 | | | | |
| 203 | PT3 | PAD17 | 8 | LOCATED | R1 C3 |
| 204 | PT2 | PAD18 | 9 | LOCATED | R1 C2 |
| 205 | PT1 | PAD19 | | LOCATED | |
| 206 | | | 10 | | R1 C1 |


208
157
PAD15
PAD0
PT1 PT2 PT3 PT4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 PT20
PL1
PL2
PL3
PL4
PR1
PR2
PR3
PR4
PAD6
CDE
SAD 0
SAD 3
SAD 4
SAD 7
BEP 0
SAD 10
SAD 17
SAD 14
SAD 15
BEP 1
SAD 16
SAD 18
SAD 19
SAD 20
SAD 21
SAD 22
SAD 23
BEP 2
PR20
PB1 PB2 PB3 PB4
PB20
53
104

POS = RXX CYY

PAR: Place And Route ORCA Foundry 9.0.

Fri Feb 28 02:40:58 1997

PAD Specification File
**********

INPUT FILE:  /nypv/pvproj/bsp301-0/bspr04.ncd
OUTPUT FILE:  /nypv/pvproj/bsp301-0/bspr04.dir/5_4.ncd
PART TYPE:  or2c15a
SPEED GRADE:  3
PACKAGE:  S208

Fri Feb 28 02:40:58 1997

Pinout by Component Name:

| Comp Name | Pin Number |
|---|---|
| AC | 78 |
| ACK | 70 |
| ACKN | 71 |
| ACN | 77 |
| ASACK | 75 |
| ASACKN | 76 |
| AUXA0 | 164 |
| AUXA1 | 163 |
| AUXA2 | 162 |
| AUXA3 | 161 |
| AUXA4 | 160 |
| AUXD0 | 153 |
| AUXD1 | 152 |
| AUXD2 | 151 |
| AUXD3 | 150 |
| AUXD4 | 149 |
| AUXD5 | 148 |
| AUXD6 | 147 |
| AUXD7 | 146 |
| BEP0 | 134 |
| BEP1 | 123 |
| BEP2 | 109 |
| BEP3 | 88 |
| BGINN | 64 |
| BGOUTN | 63 |
| BRQ | 67 |
| BUSY | 66 |
| CBE0 | 174 |
| CBE1 | 185 |
| CBE2 | 199 |
| CBE3 | 9 |
| CLK | 22 |
| DEVSELN | 193 |
| DS | 72 |
| DSN | 73 |
| ENEDRVN | 82 |

| | |
|---|---|
| ENEREC | 83 |
| ERRLEDN | 85 |
| FRAMEN | 198 |
| GA0N | 61 |
| GA1N | 60 |
| GNTN1 | 32 |
| GNTN2 | 33 |
| GNTN3 | 34 |
| GNTN4 | 35 |
| INTAN | 19 |
| INTBN | 20 |
| INTCN | 24 |
| INTDN | 25 |
| IRDYN | 197 |
| IRQ | 62 |
| OEFCNN | 154 |
| OEIDN | 159 |
| PAD0 | 165 |
| PAD1 | 166 |
| PAD10 | 178 |
| PAD11 | 179 |
| PAD12 | 180 |
| PAD13 | 181 |
| PAD14 | 183 |
| PAD15 | 184 |
| PAD16 | 201 |
| PAD17 | 203 |
| PAD18 | 204 |
| PAD19 | 205 |
| PAD2 | 167 |
| PAD20 | 5 |
| PAD21 | 6 |
| PAD22 | 7 |
| PAD23 | 8 |
| PAD24 | 10 |
| PAD25 | 11 |
| PAD26 | 13 |
| PAD27 | 14 |
| PAD28 | 15 |
| PAD29 | 16 |
| PAD3 | 169 |
| PAD30 | 17 |
| PAD31 | 18 |
| PAD4 | 170 |
| PAD5 | 171 |
| PAD6 | 172 |
| PAD7 | 173 |
| PAD8 | 175 |
| PAD9 | 176 |
| PAR | 188 |
| PAS | 80 |
| PASN | 81 |
| PERRN | 189 |
| REQ1N | 27 |
| REQ2N | 28 |
| REQ3N | 29 |

## Pin Information (continued)

**Table 21. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 208-Pin SQFP/SQFP2 Pinout**

| Pin | 2C/2T04A Pad | 2C/2T06A Pad | 2C/2T08A Pad | 2C/2T10A Pad | 2C/2T12A Pad | 2C/2T15A Pad | 2C/2T26A Pad | 2C/2T40A Pad | Function |
|---|---|---|---|---|---|---|---|---|---|
| 1 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 2 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 3 | PL1D | PL1D | PL1D | PL1D | PL1D | PL1D | PL1D | PL1D | I/O |
| 4 | PL1C | PL1A | PL2D | PL2D | PL2D | PL2D | PL2D | PL3D | I/O-A0 |
| 5 | PL1B | PL2D | PL3D | PL3D | PL3D | PL4D | PL4D | PL5D | I/O-VDD5 |
| 6 | See Note | PL2C | PL3C | PL3C | PL3A | PL4A | PL4A | PL6D | I/O |
| 7 | PL1A | PL2A | PL3A | PL3A | PL4A | PL5A | PL5A | PL8D | I/O-A1 |
| 8 | PL2D | PL3D | PL4D | PL4A | PL5A | PL6A | PL6A | PL9A | I/O-A2 |
| 9 | PL2C | PL3C | PL4C | PL5C | PL6D | PL7D | PL7D | PL10D | I/O |
| 10 | PL2B | PL3B | PL4B | PL5B | PL6B | PL7B | PL7B | PL10B | I/O |
| 11 | PL2A | PL3A | PL4A | PL5A | PL6A | PL7A | PL7A | PL10A | I/O-A3 |
| 12 | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD |
| 13 | PL3D | PL4D | PL5D | PL6D | PL7D | PL8D | PL8D | PL11D | I/O |
| 14 | PL3C | PL4C | PL5C | PL6C | PL7C | PL8C | PL8A | PL11A | I/O |
| 15 | PL3B | PL4B | PL5B | PL6B | PL7B | PL8B | PL9D | PL12D | I/O |
| 16 | PL3A | PL4A | PL5A | PL6A | PL7A | PL8A | PL9A | PL12A | I/O-A4 |
| 17 | PL4D | PL5D | PL6D | PL7D | PL8D | PL9D | PL10D | PL13D | I/O-A5 |
| 18 | PL4C | PL5C | PL6C | PL7C | PL8C | PL9C | PL10A | PL13A | I/O |
| 19 | PL4B | PL5B | PL6B | PL7B | PL8B | PL9B | PL11D | PL14D | I/O |
| 20 | PL4A | PL5A | PL6A | PL7A | PL8A | PL9A | PL11A | PL14A | I/O-A6 |
| 21 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 22 | PL5D | PL6D | PL7D | PL8D | PL9D | PL10D | PL12D | PL15D | I/O |
| 23 | PL5C | PL6C | PL7C | PL8C | PL9C | PL10C | PL12C | PL15C | I/O |
| 24 | PL5B | PL6B | PL7B | PL8B | PL9B | PL10B | PL12B | PL15B | I/O |
| 25 | PL5A | PL6A | PL7A | PL8A | PL9A | PL10A | PL12A | PL15A | I/O-A7 |
| 26 | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD |
| 27 | PL6D | PL7D | PL8D | PL9D | PL10D | PL11D | PL13D | PL16D | I/O |
| 28 | PL6C | PL7C | PL8C | PL9C | PL10C | PL11C | PL13C | PL16C | I/O-VDD5 |
| 29 | PL6B | PL7B | PL8B | PL9B | PL10B | PL11B | PL13B | PL16B | I/O |
| 30 | PL6A | PL7A | PL8A | PL9A | PL10A | PL11A | PL13A | PL16A | I/O-A8 |
| 31 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 32 | PL7D | PL8D | PL9D | PL10D | PL11D | PL12D | PL14D | PL17D | I/O-A9 |
| 33 | PL7C | PL8C | PL9C | PL10C | PL11C | PL12C | PL14A | PL17A | I/O |
| 34 | PL7B | PL8B | PL9B | PL10B | PL11B | PL12B | PL15D | PL18D | I/O |
| 35 | PL7A | PL8A | PL9A | PL10A | PL11A | PL12A | PL15A | PL18A | I/O-A10 |
| 36 | PL8D | PL9D | PL10D | PL11D | PL12D | PL13D | PL16D | PL19D | I/O |
| 37 | PL8C | PL9C | PL10C | PL11C | PL12C | PL13C | PL16A | PL19A | I/O |
| 38 | PL8B | PL9B | PL10B | PL11B | PL12B | PL13B | PL17D | PL20D | I/O |

Notes:
The OR2C04A and OR2T04A do not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

## Pin Information (continued)

**Table 21. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 208-Pin SQFP/SQFP2 Pinout** (continued)

| Pin | 2C/2T04A Pad | 2C/2T06A Pad | 2C/2T08A Pad | 2C/2T10A Pad | 2C/2T12A Pad | 2C/2T15A Pad | 2C/2T26A Pad | 2C/2T40A Pad | Function |
|---|---|---|---|---|---|---|---|---|---|
| 39 | PL8A | PL9A | PL10A | PL11A | PL12A | PL13A | PL17A | PL20A | I/O-A11 |
| 40 | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD |
| 41 | PL9D | PL10D | PL11D | PL12D | PL13D | PL14D | PL18D | PL21D | I/O-A12 |
| 42 | PL9C | PL10C | PL11C | PL12C | PL13B | PL14B | PL18B | PL21B | I/O |
| 43 | PL9B | PL10B | PL11B | PL12B | PL14D | PL15D | PL19D | PL22D | I/O |
| 44 | PL9A | PL10A | PL11A | PL13D | PL14B | PL15B | PL19B | PL22B | I/O-A13 |
| 45 | See Note | PL11D | PL12D | PL13B | PL15D | PL16D | PL20D | PL23D | I/O |
| 46 | PL10D | PL11A | PL12A | PL14C | PL16D | PL17D | PL21D | PL25A | I/O-A14 |
| 47 | See Note | PL12D | PL13D | PL15D | PL17D | PL18D | PL22D | PL27D | I/O |
| 48 | PL10C | PL12C | PL13A | PL15A | PL17A | PL19D | PL23D | PL28D | I/O |
| 49 | PL10B | PL12B | PL14D | PL16D | PL18C | PL19A | PL23A | PL28A | I/O |
| 50 | PL10A | PL12A | PL14A | PL16A | PL18A | PL20A | PL24A | PL30A | I/O-A15 |
| 51 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 52 | CCLK | CCLK | CCLK | CCLK | CCLK | CCLK | CCLK | CCLK | CCLK |
| 53 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 54 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 55 | PB1A | PB1A | PB1A | PB1A | PB1A | PB1A | PB1A | PB1A | I/O-A16 |
| 56 | See Note | PB1B | PB1D | PB1D | PB1D | PB2A | PB2A | PB3A | I/O |
| 57 | PB1B | PB1C | PB2A | PB2A | PB2A | PB2D | PB2D | PB3D | I/O-VDD5 |
| 58 | PB1C | PB1D | PB2D | PB2D | PB2D | PB3D | PB3D | PB4D | I/O |
| 59 | PB1D | PB2A | PB3A | PB3B | PB3D | PB4D | PB4D | PB5D | I/O-A17 |
| 60 | See Note | PB2D | PB3D | PB4D | PB4D | PB5D | PB5D | PB6D | I/O |
| 61 | PB2A | PB3A | PB4A | PB5A | PB5B | PB6B | PB6B | PB7D | I/O |
| 62 | PB2B | PB3B | PB4B | PB5B | PB5D | PB6D | PB6D | PB8D | I/O |
| 63 | PB2C | PB3C | PB4C | PB5C | PB6B | PB7B | PB7B | PB9D | I/O |
| 64 | PB2D | PB3D | PB4D | PB5D | PB6D | PB7D | PB7D | PB10D | I/O |
| 65 | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD |
| 66 | PB3A | PB4A | PB5A | PB6A | PB7A | PB8A | PB8A | PB11A | I/O |
| 67 | PB3B | PB4B | PB5B | PB6B | PB7B | PB8B | PB8D | PB11D | I/O |
| 68 | PB3C | PB4C | PB5C | PB6C | PB7C | PB8C | PB9A | PB12A | I/O |
| 69 | PB3D | PB4D | PB5D | PB6D | PB7D | PB8D | PB9D | PB12D | I/O |
| 70 | PB4A | PB5A | PB6A | PB7A | PB8A | PB9A | PB10A | PB13A | I/O |
| 71 | PB4B | PB5B | PB6B | PB7B | PB8B | PB9B | PB10D | PB13D | I/O |
| 72 | PB4C | PB5C | PB6C | PB7C | PB8C | PB9C | PB11A | PB14A | I/O |
| 73 | PB4D | PB5D | PB6D | PB7D | PB8D | PB9D | PB11D | PB14D | I/O |
| 74 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |

Notes:
The OR2C04A and OR2T04A do not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

## Pin Information (continued)

**Table 21. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 208-Pin SQFP/SQFP2 Pinout** (continued)

| Pin | 2C/2T04A Pad | 2C/2T06A Pad | 2C/2T08A Pad | 2C/2T10A Pad | 2C/2T12A Pad | 2C/2T15A Pad | 2C/2T26A Pad | 2C/2T40A Pad | Function |
|---|---|---|---|---|---|---|---|---|---|
| 75 | PB5A | PB6A | PB7A | PB8A | PB9A | PB10A | PB12A | PB15A | I/O |
| 76 | PB5B | PB6B | PB7B | PB8B | PB9B | PB10B | PB12B | PB15B | I/O |
| 77 | PB5C | PB6C | PB7C | PB8C | PB9C | PB10C | PB12C | PB15C | I/O |
| 78 | PB5D | PB6D | PB7D | PB8D | PB9D | PB10D | PB12D | PB15D | I/O |
| 79 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 80 | PB6A | PB7A | PB8A | PB9A | PB10A | PB11A | PB13A | PB16A | I/O |
| 81 | PB6B | PB7B | PB8B | PB9B | PB10B | PB11B | PB13B | PB16B | I/O |
| 82 | PB6C | PB7C | PB8C | PB9C | PB10C | PB11C | PB13C | PB16C | I/O |
| 83 | PB6D | PB7D | PB8D | PB9D | PB10D | PB11D | PB13D | PB16D | I/O |
| 84 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 85 | PB7A | PB8A | PB9A | PB10A | PB11A | PB12A | PB14A | PB17A | I/O-VDD5 |
| 86 | PB7B | PB8B | PB9B | PB10B | PB11B | PB12B | PB14D | PB17D | I/O |
| 87 | PB7C | PB8C | PB9C | PB10C | PB11C | PB12C | PB15A | PB18A | I/O |
| 88 | PB7D | PB8D | PB9D | PB10D | PB11D | PB12D | PB15D | PB18D | I/O |
| 89 | PB8A | PB9A | PB10A | PB11A | PB12A | PB13A | PB16A | PB19A | I/O-HDC |
| 90 | PB8B | PB9B | PB10B | PB11B | PB12B | PB13B | PB16D | PB19D | I/O |
| 91 | PB8C | PB9C | PB10C | PB11C | PB12C | PB13C | PB17A | PB20A | I/O |
| 92 | PB8D | PB9D | PB10D | PB11D | PB12D | PB13D | PB17D | PB20D | I/O |
| 93 | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD |
| 94 | PB9A | PB10A | PB11A | PB12A | PB13A | PB14A | PB18A | PB21A | I/O-$\overline{\text{LDC}}$ |
| 95 | PB9B | PB10B | PB11D | PB13A | PB13D | PB14D | PB18D | PB22D | I/O |
| 96 | PB9C | PB10C | PB12A | PB13B | PB14A | PB15A | PB19A | PB23A | I/O |
| 97 | PB9D | PB10D | PB12B | PB13C | PB14D | PB15D | PB19D | PB24D | I/O |
| 98 | PB10A | PB11A | PB12C | PB13D | PB15A | PB16A | PB20A | PB25A | I/O-$\overline{\text{INIT}}$ |
| 99 | PB10B | PB11C | PB12D | PB14A | PB16A | PB17A | PB21A | PB26A | I/O |
| 100 | PB10C | PB11D | PB13A | PB15A | PB17A | PB18A | PB22A | PB27A | I/O |
| 101 | PB10D | PB12A | PB13D | PB15D | PB18A | PB19D | PB23D | PB28D | I/O |
| 102 | See Note | PB12D | PB14D | PB16D | PB18D | PB20D | PB24D | PB30D | I/O |
| 103 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 104 | DONE | DONE | DONE | DONE | DONE | DONE | DONE | DONE | DONE |
| 105 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 106 | $\overline{\text{RESET}}$ | $\overline{\text{RESET}}$ | $\overline{\text{RESET}}$ | $\overline{\text{RESET}}$ | $\overline{\text{RESET}}$ | $\overline{\text{RESET}}$ | $\overline{\text{RESET}}$ | $\overline{\text{RESET}}$ | $\overline{\text{RESET}}$ |
| 107 | $\overline{\text{PRGM}}$ | $\overline{\text{PRGM}}$ | $\overline{\text{PRGM}}$ | $\overline{\text{PRGM}}$ | $\overline{\text{PRGM}}$ | $\overline{\text{PRGM}}$ | $\overline{\text{PRGM}}$ | $\overline{\text{PRGM}}$ | $\overline{\text{PRGM}}$ |
| 108 | PR10A | PR12A | PR14A | PR16A | PR18A | PR20A | PR24A | PR30A | I/O-M0 |
| 109 | PR10B | PR12D | PR13A | PR15A | PR18D | PR19A | PR23A | PR28A | I/O |
| 110 | PR10C | PR11A | PR13D | PR15D | PR17B | PR18A | PR22A | PR27A | I/O |

Notes:
The OR2C04A and OR2T04A do not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

## Pin Information (continued)

**Table 21. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 208-Pin SQFP/SQFP2 Pinout** (continued)

| Pin | 2C/2T04A Pad | 2C/2T06A Pad | 2C/2T08A Pad | 2C/2T10A Pad | 2C/2T12A Pad | 2C/2T15A Pad | 2C/2T26A Pad | 2C/2T40A Pad | Function |
|---|---|---|---|---|---|---|---|---|---|
| 111 | PR10D | PR11B | PR12A | PR14A | PR16A | PR17A | PR21A | PR26A | I/O |
| 112 | PR9A | PR10A | PR11A | PR13B | PR15D | PR16D | PR20D | PR23D | I/O-M1 |
| 113 | PR9B | PR10B | PR11B | PR13C | PR14A | PR15A | PR19A | PR22A | I/O |
| 114 | PR9C | PR10C | PR11C | PR12A | PR14D | PR15D | PR19D | PR22D | I/O-VDD5 |
| 115 | PR9D | PR10D | PR11D | PR12B | PR13A | PR14A | PR18A | PR21A | I/O |
| 116 | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD |
| 117 | PR8A | PR9A | PR10A | PR11A | PR12A | PR13A | PR17A | PR20A | I/O-M2 |
| 118 | PR8B | PR9B | PR10B | PR11B | PR12B | PR13B | PR17D | PR20D | I/O |
| 119 | PR8C | PR9C | PR10C | PR11C | PR12C | PR13C | PR16A | PR19A | I/O |
| 120 | PR8D | PR9D | PR10D | PR11D | PR12D | PR13D | PR16D | PR19D | I/O |
| 121 | PR7A | PR8A | PR9A | PR10A | PR11A | PR12A | PR15A | PR18A | I/O-M3 |
| 122 | PR7B | PR8B | PR9B | PR10B | PR11B | PR12B | PR15D | PR18D | I/O |
| 123 | PR7C | PR8C | PR9C | PR10C | PR11C | PR12C | PR14A | PR17A | I/O |
| 124 | PR7D | PR8D | PR9D | PR10D | PR11D | PR12D | PR14D | PR17D | I/O |
| 125 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 126 | PR6A | PR7A | PR8A | PR9A | PR10A | PR11A | PR13A | PR16A | I/O |
| 127 | PR6B | PR7B | PR8B | PR9B | PR10B | PR11B | PR13B | PR16B | I/O |
| 128 | PR6C | PR7C | PR8C | PR9C | PR10C | PR11C | PR13C | PR16C | I/O |
| 129 | PR6D | PR7D | PR8D | PR9D | PR10D | PR11D | PR13D | PR16D | I/O |
| 130 | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD |
| 131 | PR5A | PR6A | PR7A | PR8A | PR9A | PR10A | PR12A | PR15A | I/O |
| 132 | PR5B | PR6B | PR7B | PR8B | PR9B | PR10B | PR12B | PR15B | I/O |
| 133 | PR5C | PR6C | PR7C | PR8C | PR9C | PR10C | PR12C | PR15C | I/O |
| 134 | PR5D | PR6D | PR7D | PR8D | PR9D | PR10D | PR12D | PR15D | I/O |
| 135 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 136 | PR4A | PR5A | PR6A | PR7A | PR8A | PR9A | PR11A | PR14A | I/O-VDD5 |
| 137 | PR4B | PR5B | PR6B | PR7B | PR8B | PR9B | PR11D | PR14D | I/O |
| 138 | PR4C | PR5C | PR6C | PR7C | PR8C | PR9C | PR10A | PR13A | I/O |
| 139 | PR4D | PR5D | PR6D | PR7D | PR8D | PR9D | PR10D | PR13D | I/O |
| 140 | PR3A | PR4A | PR5A | PR6A | PR7A | PR8A | PR9A | PR12A | I/O-CS1 |
| 141 | PR3B | PR4B | PR5B | PR6B | PR7B | PR8B | PR9D | PR12D | I/O |
| 142 | PR3C | PR4C | PR5C | PR6C | PR7C | PR8C | PR8A | PR11A | I/O |
| 143 | PR3D | PR4D | PR5D | PR6D | PR7D | PR8D | PR8D | PR11D | I/O |
| 144 | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD |
| 145 | PR2A | PR3A | PR4A | PR5A | PR6A | PR7A | PR7A | PR10A | I/O-$\overline{CS0}$ |
| 146 | PR2B | PR3B | PR4B | PR4B | PR6B | PR7B | PR7B | PR10B | I/O |

Notes:
The OR2C04A and OR2T04A do not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

## Pin Information (continued)

**Table 21. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 208-Pin SQFP/SQFP2 Pinout** (continued)

| Pin | 2C/2T04A Pad | 2C/2T06A Pad | 2C/2T08A Pad | 2C/2T10A Pad | 2C/2T12A Pad | 2C/2T15A Pad | 2C/2T26A Pad | 2C/2T40A Pad | Function |
|---|---|---|---|---|---|---|---|---|---|
| 147 | PR2C | PR3C | PR4C | PR4C | PR5B | PR6B | PR6B | PR9B | I/O |
| 148 | PR2D | PR3D | PR4D | PR4D | PR5D | PR6D | PR6D | PR9D | I/O |
| 149 | PR1A | PR2A | PR3A | PR3A | PR4A | PR5A | PR5A | PR8A | I/O-$\overline{RD}$ |
| 150 | PR1B | PR2C | PR3C | PR3C | PR4D | PR5D | PR5D | PR6A | I/O |
| 151 | PR1C | PR2D | PR3D | PR3D | PR3A | PR4A | PR4A | PR5A | I/O |
| 152 | PR1D | PR1A | PR2A | PR2A | PR2A | PR3A | PR3A | PR4A | I/O-$\overline{WR}$ |
| 153 | See Note | PR1C | PR2D | PR2D | PR2C | PR2A | PR2A | PR3A | I/O |
| 154 | See Note | PR1D | PR1A | PR1A | PR1A | PR1A | PR1A | PR2A | I/O |
| 155 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 156 | $\overline{RD\_CFGN}$ | $\overline{RD\_CFGN}$ | $\overline{RD\_CFGN}$ | $\overline{RD\_CFGN}$ | $\overline{RD\_CFGN}$ | $\overline{RD\_CFGN}$ | $\overline{RD\_CFGN}$ | $\overline{RD\_CFGN}$ | $\overline{RD\_CFGN}$ |
| 157 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 158 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 159 | PT10D | PT12D | PT14D | PT16D | PT18D | PT20D | PT24D | PT30D | I/O |
| 160 | PT10C | PT12A | PT13D | PT15D | PT17D | PT19A | PT23A | PT28A | I/O-RDY/RCLK |
| 161 | PT10B | PT11D | PT13A | PT15A | PT16D | PT17D | PT21D | PT26D | I/O |
| 162 | PT10A | PT11C | PT12D | PT14D | PT16A | PT17A | PT21A | PT26A | I/O |
| 163 | PT9D | PT11A | PT12C | PT13D | PT15D | PT16D | PT20D | PT25D | I/O-D7 |
| 164 | PT9C | PT10D | PT12A | PT13B | PT14D | PT15D | PT19D | PT24D | I/O-VDD5 |
| 165 | PT9B | PT10C | PT11D | PT13A | PT14A | PT15A | PT19A | PT23D | I/O |
| 166 | See Note | PT10B | PT11C | PT12D | PT13D | PT14D | PT18D | PT22D | I/O |
| 167 | PT9A | PT10A | PT11B | PT12B | PT13B | PT14B | PT18B | PT21D | I/O-D6 |
| 168 | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD |
| 169 | PT8D | PT9D | PT10D | PT11D | PT12D | PT13D | PT17D | PT20D | I/O |
| 170 | PT8C | PT9C | PT10C | PT11C | PT12C | PT13C | PT17A | PT20A | I/O |
| 171 | PT8B | PT9B | PT10B | PT11B | PT12B | PT13B | PT16D | PT19D | I/O |
| 172 | PT8A | PT9A | PT10A | PT11A | PT12A | PT13A | PT16A | PT19A | I/O-D5 |
| 173 | PT7D | PT8D | PT9D | PT10D | PT11D | PT12D | PT15D | PT18D | I/O |
| 174 | PT7C | PT8C | PT9C | PT10C | PT11C | PT12C | PT15A | PT18A | I/O |
| 175 | PT7B | PT8B | PT9B | PT10B | PT11B | PT12B | PT14D | PT17D | I/O |
| 176 | PT7A | PT8A | PT9A | PT10A | PT11A | PT12A | PT14A | PT17A | I/O-D4 |
| 177 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 178 | PT6D | PT7D | PT8D | PT9D | PT10D | PT11D | PT13D | PT16D | I/O |
| 179 | PT6C | PT7C | PT8C | PT9C | PT10C | PT11C | PT13C | PT16C | I/O |
| 180 | PT6B | PT7B | PT8B | PT9B | PT10B | PT11B | PT13B | PT16B | I/O |
| 181 | PT6A | PT7A | PT8A | PT9A | PT10A | PT11A | PT13A | PT16A | I/O-D3 |
| 182 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |

Notes:
The OR2C04A and OR2T04A do not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

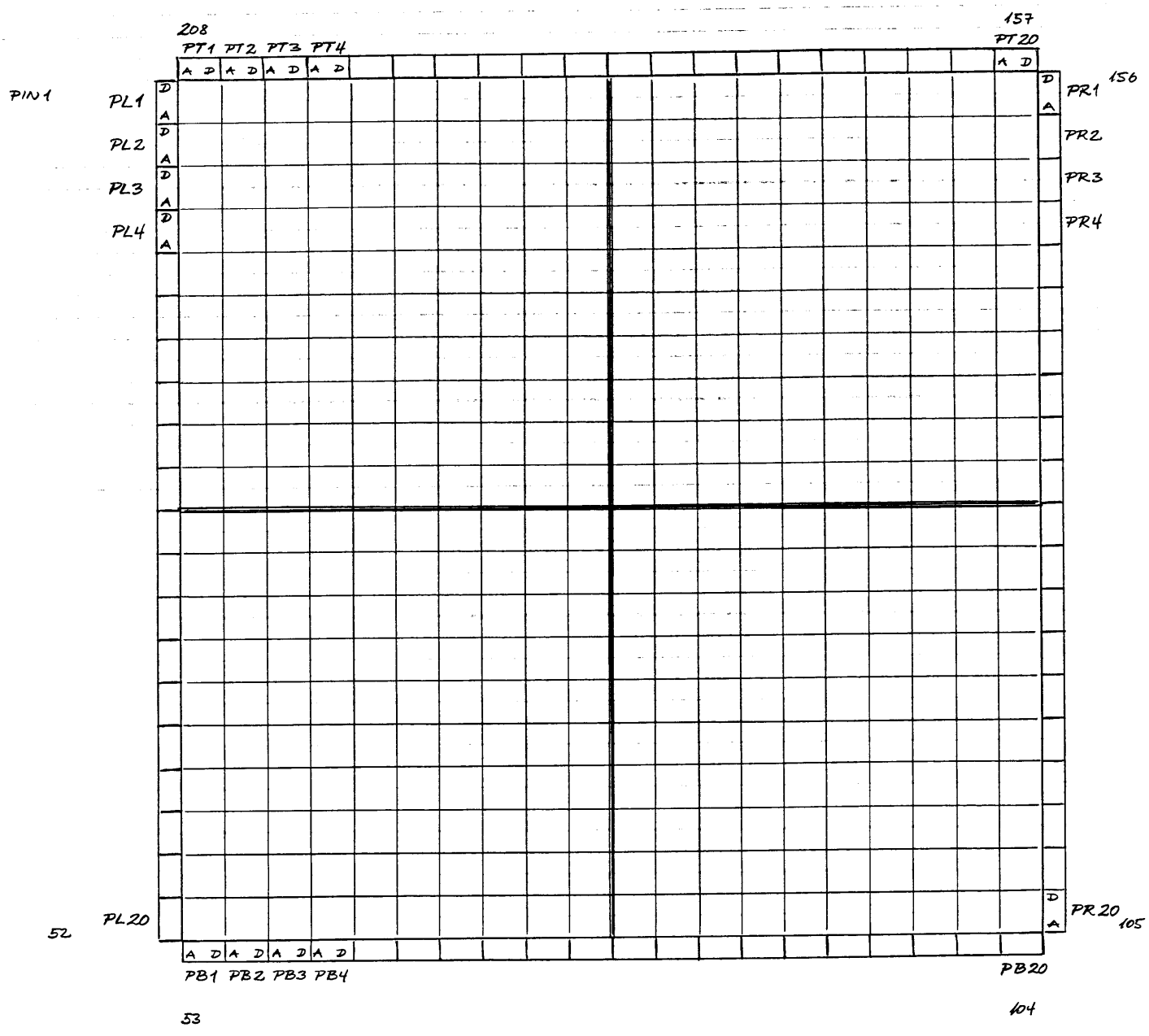
## Pin Information (continued)

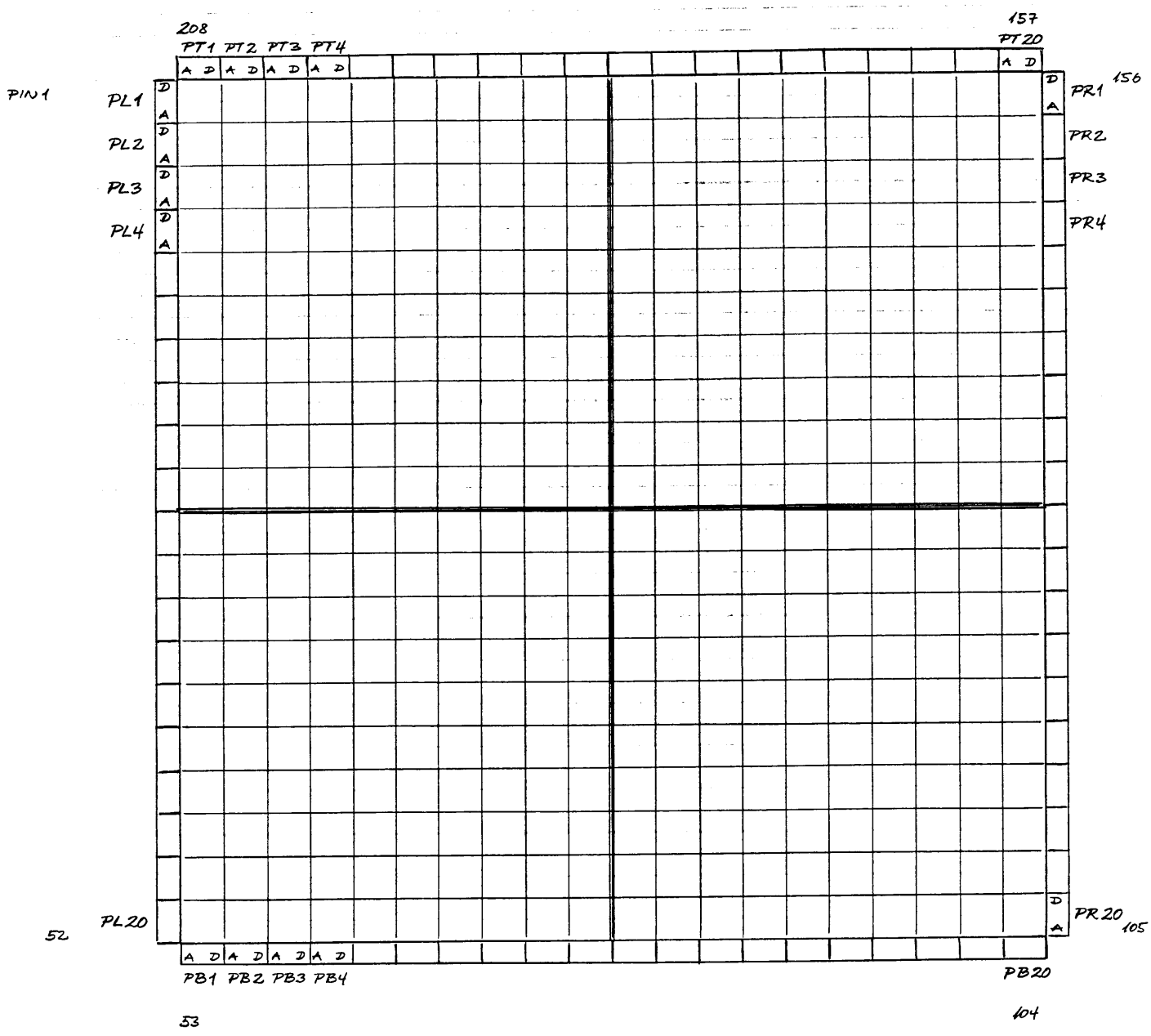
**Table 21. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 208-Pin SQFP/SQFP2 Pinout** (continued)

| Pin | 2C/2T04A Pad | 2C/2T06A Pad | 2C/2T08A Pad | 2C/2T10A Pad | 2C/2T12A Pad | 2C/2T15A Pad | 2C/2T26A Pad | 2C/2T40A Pad | Function |
|---|---|---|---|---|---|---|---|---|---|
| 183 | PT5D | PT6D | PT7D | PT8D | PT9D | PT10D | PT12D | PT15D | I/O |
| 184 | PT5C | PT6C | PT7C | PT8C | PT9C | PT10C | PT12C | PT15C | I/O |
| 185 | PT5B | PT6B | PT7B | PT8B | PT9B | PT10B | PT12B | PT15B | I/O-VDD5 |
| 186 | PT5A | PT6A | PT7A | PT8A | PT9A | PT10A | PT12A | PT15A | I/O-D2 |
| 187 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 188 | PT4D | PT5D | PT6D | PT7D | PT8D | PT9D | PT11D | PT14D | I/O-D1 |
| 189 | PT4C | PT5C | PT6C | PT7C | PT8C | PT9C | PT11A | PT14A | I/O |
| 190 | PT4B | PT5B | PT6B | PT7B | PT8B | PT9B | PT10D | PT13D | I/O |
| 191 | PT4A | PT5A | PT6A | PT7A | PT8A | PT9A | PT10A | PT13A | I/O-D0/DIN |
| 192 | PT3D | PT4D | PT5D | PT6D | PT7D | PT8D | PT9D | PT12D | I/O |
| 193 | PT3C | PT4C | PT5C | PT6C | PT7C | PT8C | PT9A | PT12A | I/O |
| 194 | PT3B | PT4B | PT5B | PT6B | PT7B | PT8B | PT8D | PT11D | I/O |
| 195 | PT3A | PT4A | PT5A | PT6A | PT7A | PT8A | PT8A | PT11A | I/O-DOUT |
| 196 | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD | VDD |
| 197 | PT2D | PT3D | PT4D | PT5D | PT6D | PT7D | PT7D | PT10D | I/O |
| 198 | PT2C | PT3C | PT4C | PT5A | PT6A | PT7A | PT7A | PT9A | I/O |
| 199 | PT2B | PT3B | PT4B | PT4D | PT5C | PT6C | PT6C | PT8A | I/O |
| 200 | PT2A | PT3A | PT4A | PT4A | PT5A | PT6A | PT6A | PT7A | I/O-TDI |
| 201 | See Note | PT2D | PT3D | PT3D | PT4A | PT5A | PT5A | PT6A | I/O |
| 202 | PT1D | PT2A | PT3A | PT3A | PT3A | PT4A | PT4A | PT5A | I/O-TMS |
| 203 | See Note | PT1D | PT2D | PT2D | PT2C | PT3A | PT3A | PT4A | I/O |
| 204 | PT1C | PT1C | PT2A | PT2A | PT2A | PT2A | PT2A | PT3A | I/O |
| 205 | PT1B | PT1B | PT1D | PT1D | PT1D | PT1D | PT1D | PT2D | I/O |
| 206 | PT1A | PT1A | PT1A | PT1A | PT1A | PT1A | PT1A | PT1A | I/O-TCK |
| 207 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| 208 | RD_DATA/ TDO | RD_DATA/ TDO | RD_DATA/ TDO | RD_DATA/ TDO | RD_DATA/ TDO | RD_DATA/ TDO | RD_DATA/ TDO | RD_DATA/ TDO | RD_DATA/TDO |

Notes:
The OR2C04A and OR2T04A do not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.


208
157
PT1 PT2 PT3 PT4
PT20
A D
PIN 1
PL1
PL2
PL3
PL4
PL20
PR1
PR2
PR3
PR4
PR20
156
105
52
PB1 PB2 PB3 PB4
PB20
53
104


208
PT1 PT2 PT3 PT4
157
PT20
A D
PIN 1
PL1
PL2
PL3
PL4
PL20
PR1
PR2
PR3
PR4
PR20
156
105
52
PB1 PB2 PB3 PB4
PB20
53
104

# 6 REGISTERS

## 6.1 Register Summary

The base address for BAIO access to a submodule depends on the submodule position as listed below.

Position 1: 0x0800 0000
Position 2: 0x0a00 0000
Position 3: 0x0c00 0000

Register addresses are indicated as offsets to the base addresses.

| Address | Register Name |
|---|---|
| 0x0003 | Submodule ID, 32 bytes in 32 words |
| 0x0083 | Submodule Revision, 32 bytes in 32 words |
| 0x0103 | SMI Status Register |
| 0x0183 | SMI Control Register: |
| 0x0200 | PCI Status Register |
| 0x0280 | PCI Control Register |
| 0x0300 | PCI Window Base 1 Register (PWBR1) |
| 0x0380 | PCI Window Base 2 Register (PWBR2) |
| 0x0400 | PCI Master Latency Timer Register (PMLTR) |

## 1 PIN ASSIGNMENTS

| Pin | Pad | Signal | Input PLC | Output PLC |
|---|---|---|---|---|
| 1 | VSS | | ..... | ..... |
| 2 | VSS | | ..... | ..... |
| 3 | PL1D | ..... | ..... | ..... |
| 4 | PL2D | ..... | ..... | ..... |
| 5 | PL4D | PAD20 | ..... | ..... |
| 6 | A | ....21 | ..... | ..... |
| 7 | PL5A | ....22 | ..... | ..... |
| 8 | PL6A | ....23 | ..... | ..... |
| 9 | PL7D | CBE3 | ..... | ..... |
| 10 | B | PAD24 | ..... | ..... |
| 11 | A | ....25 | ..... | ..... |
| 12 | VDD | | ..... | ..... |
| 13 | PL8D | PAD26 | ..... | ..... |
| 14 | C | ....27 | ..... | ..... |
| 15 | B | ....28 | ..... | ..... |
| 16 | A | ....29 | ..... | ..... |
| 17 | PL9D | ....30 | ..... | ..... |
| 18 | C | ....31 | ..... | ..... |
| 19 | B | INTAN | ..... | ..... |
| 20 | A | ...BN | ..... | ..... |
| 21 | VSS | | ..... | ..... |
| 22 | PL10D | CLK | ..... | ..... |
| 23 | C | ..... | ..... | ..... |
| 24 | B | INTCN | ..... | ..... |
| 25 | A | ...DN | ..... | ..... |
| 26 | VDD | ..... | ..... | ..... |
| 27 | PL11D | REQIN | ..... | ..... |
| 28 | C | ....2N | ..... | ..... |
| 29 | B | ....3N | ..... | ..... |
| 30 | A | ....4N | ..... | ..... |
| 31 | VSS | | ..... | ..... |
| 32 | PL12D | GNT1N | ..... | ..... |
| 33 | C | ....2N | ..... | ..... |
| 34 | B | ....3N | ..... | ..... |
| 35 | A | ....4N | ..... | ..... |
| 36 | PL13D | ..... | ..... | ..... |
| 37 | C | ..... | ..... | ..... |
| 38 | B | ..... | ..... | ..... |
| 39 | A | ..... | ..... | ..... |
| 40 | VDD | | ..... | ..... |
| 41 | PL14D | ..... | ..... | ..... |
| 42 | B | ..... | ..... | ..... |
| 43 | PL15D | ..... | ..... | ..... |
| 44 | B | ..... | ..... | ..... |
| 45 | PL16D | ..... | ..... | ..... |
| 46 | PL17D | ..... | ..... | ..... |
| 47 | PL18D | ..... | ..... | ..... |
| 48 | PL19D | ..... | ..... | ..... |
| 49 | A | ..... | ..... | ..... |
| 50 | PL20A | ..... | ..... | ..... |
| 51 | VSS | | ..... | ..... |
| 52 | CCLK | | ..... | ..... |

| Pin | Pad | Signal | Input PLC | Output PLC |
|---|---|---|---|---|
| 53 | VSS | | ..... | ..... |
| 54 | VSS | | ..... | ..... |
| 55 | PB1A | ..... | ..... | ..... |
| 56 | PB2A | ..... | ..... | ..... |
| 57 | D | ..... | ..... | ..... |
| 58 | PB3D | ..... | ..... | ..... |
| 59 | PB4D | ..... | ..... | ..... |
| 60 | PB5D | GAIN | ..... | ..... |
| 61 | PB6B | GAON | ..... | ..... |
| 62 | D | IRQ | ..... | ..... |
| 63 | PB7B | BGOUTN | ..... | ..... |
| 64 | D | BGINN | ..... | ..... |
| 65 | VDD | | ..... | ..... |
| 66 | PB8A | BUSY | ..... | ..... |
| 67 | B | BRQ | ..... | ..... |
| 68 | C | WR | ..... | ..... |
| 69 | D | WRN | ..... | ..... |
| 70 | PB9A | ACK | ..... | ..... |
| 71 | B | ACKN | ..... | ..... |
| 72 | C | DS | ..... | ..... |
| 73 | D | DSN | ..... | ..... |
| 74 | VSS | | ..... | ..... |
| 75 | PB10A | ASACK | ..... | ..... |
| 76 | B | ASACKN | ..... | ..... |
| 77 | C | ACN | ..... | ..... |
| 78 | D | AC | ..... | ..... |
| 79 | VSS | | ..... | ..... |
| 80 | PB11A | PAS | ..... | ..... |
| 81 | B | PASN | ..... | ..... |
| 82 | B | ENEDRUN | ..... | ..... |
| 83 | D | ENEREC | ..... | ..... |
| 84 | VSS | | ..... | ..... |
| 85 | PB12A | ERRLEDN | ..... | ..... |
| 86 | B | ..... | ..... | ..... |
| 87 | C | ..... | ..... | ..... |
| 88 | D | BEP3 | ..... | ..... |
| 89 | HDC | HDC | ..... | ..... |
| 90 | PB13B | SAD31 | ..... | ..... |
| 91 | C | ....30 | ..... | ..... |
| 92 | D | ....29 | ..... | ..... |
| 93 | VDD | | ..... | ..... |
| 94 | LDC | LDC | ..... | ..... |
| 95 | PB14D | SAD28 | ..... | ..... |
| 96 | PB15A | ....27 | ..... | ..... |
| 97 | D | ....26 | ..... | ..... |
| 98 | INIT | INIT | ..... | ..... |
| 99 | PB17A | SAD25 | ..... | ..... |
| 100 | PB18A | ....24 | ..... | ..... |
| 101 | PB19D | ..... | ..... | ..... |
| 102 | PB20D | ..... | ..... | ..... |
| 103 | VSS | | ..... | ..... |
| 104 | DONE | | ..... | ..... |

| Pin | Pad | Signal | Input PLC | Output PLC |
|---|---|---|---|---|
| 105 | VSS | | ..... | ..... |
| 106 | RESET | RESET | ..... | ..... |
| 107 | PRGM | ..... | ..... | ..... |
| 108 | M0 | | ..... | ..... |
| 109 | PR19A | ..... | ..... | ..... |
| 110 | PR18A | ..... | ..... | ..... |
| 111 | PR17A | ..... | ..... | ..... |
| 112 | M1 | | ..... | ..... |
| 113 | PR15A | ..... | ..... | ..... |
| 114 | D | ..... | ..... | ..... |
| 115 | PR14A | ..... | ..... | ..... |
| 116 | VDD | | ..... | ..... |
| 117 | M2 | | ..... | ..... |
| 118 | PR13B | ..... | ..... | ..... |
| 119 | C | ..... | ..... | ..... |
| 120 | D | ..... | ..... | ..... |
| 121 | M3 | | ..... | ..... |
| 122 | PR12B | ..... | ..... | ..... |
| 123 | C | ..... | ..... | ..... |
| 124 | D | ..... | ..... | ..... |
| 125 | VSS | | ..... | ..... |
| 126 | PR11A | ..... | ..... | ..... |
| 127 | B | ..... | ..... | ..... |
| 128 | C | ..... | ..... | ..... |
| 129 | D | ..... | ..... | ..... |
| 130 | VDD | | ..... | ..... |
| 131 | PR10A | ..... | ..... | ..... |
| 132 | B | ..... | ..... | ..... |
| 133 | C | ..... | ..... | ..... |
| 134 | D | ..... | ..... | ..... |
| 135 | VSS | | ..... | ..... |
| 136 | PR9A | ..... | ..... | ..... |
| 137 | B | ..... | ..... | ..... |
| 138 | C | ..... | ..... | ..... |
| 139 | D | ..... | ..... | ..... |
| 140 | PR8A | ..... | ..... | ..... |
| 141 | B | ..... | ..... | ..... |
| 142 | C | ..... | ..... | ..... |
| 143 | D | ..... | ..... | ..... |
| 144 | VDD | | ..... | ..... |
| 145 | PR7A | ..... | ..... | ..... |
| 146 | B | ..... | ..... | ..... |
| 147 | PR6B | ..... | ..... | ..... |
| 148 | D | ..... | ..... | ..... |
| 149 | PR5A | ..... | ..... | ..... |
| 150 | D | ..... | ..... | ..... |
| 151 | PR4A | ..... | ..... | ..... |
| 152 | PR3A | ..... | ..... | ..... |
| 153 | PR2A | ..... | ..... | ..... |
| 154 | PR1A | ..... | ..... | ..... |
| 155 | VSS | | ..... | ..... |
| 156 | RDCFN | | ..... | ..... |

| Pin | Pad | Signal | Input PLC | Output PLC |
|---|---|---|---|---|
| 157 | VSS | | ..... | ..... |
| 158 | VSS | | ..... | ..... |
| 159 | PT20D | ..... | ..... | ..... |
| 160 | PT19A | ..... | ..... | ..... |
| 161 | PT17D | ..... | ..... | ..... |
| 162 | A | ..... | ..... | ..... |
| 163 | PT16D | ..... | ..... | ..... |
| 164 | PT15D | ..... | ..... | ..... |
| 165 | A | ..... | ..... | ..... |
| 166 | PT14D | ..... | ..... | ..... |
| 167 | B | ..... | ..... | ..... |
| 168 | VDD | | ..... | ..... |
| 169 | PT13D | ..... | ..... | ..... |
| 170 | C | ..... | ..... | ..... |
| 171 | B | ..... | ..... | ..... |
| 172 | A | ..... | ..... | ..... |
| 173 | PT12D | ..... | ..... | ..... |
| 174 | C | ..... | ..... | ..... |
| 175 | B | ..... | ..... | ..... |
| 176 | PT13A | ..... | ..... | ..... |
| 177 | VSS | | ..... | ..... |
| 178 | PT11D | ..... | ..... | ..... |
| 179 | C | ..... | ..... | ..... |
| 180 | B | ..... | ..... | ..... |
| 181 | A | ..... | ..... | ..... |
| 182 | VSS | | ..... | ..... |
| 183 | PT10D | ..... | ..... | ..... |
| 184 | C | ..... | ..... | ..... |
| 185 | B | ..... | ..... | ..... |
| 186 | A | ..... | ..... | ..... |
| 187 | VSS | | ..... | ..... |
| 188 | PT9D | PAR | ..... | ..... |
| 189 | C | PERRN | ..... | ..... |
| 190 | B | SERRN | ..... | ..... |
| 191 | DIN | DIN | ..... | ..... |
| 192 | PT8D | STOPN | ..... | ..... |
| 193 | C | DEVSELN | ..... | ..... |
| 194 | B | TRDYN | ..... | ..... |
| 195 | DOUT | DOUT | ..... | ..... |
| 196 | VDD | | ..... | ..... |
| 197 | PT7D | IRDYN | ..... | ..... |
| 198 | A | FRAMEN | ..... | ..... |
| 199 | PT6C | CBE2 | ..... | ..... |
| 200 | A | TDI | ..... | ..... |
| 201 | PT5A | PAD16 | ..... | ..... |
| 202 | PT4A | TMS | ..... | ..... |
| 203 | PT3A | PAD17 | ..... | ..... |
| 204 | PT2A | ....18 | ..... | ..... |
| 205 | PT1D | ....19 | ..... | ..... |
| 206 | A | TCK | ..... | ..... |
| 207 | VSS | | ..... | ..... |
| 208 | RDDAT | | ..... | ..... |

**DESIGN NOTE**

| | |
|---|---|
| Name: | BSP301 Pin Assignment |
| Date: | 1997-02-27 |
| Author: | aaj |
| Version: | 1.1 |
| Document ID: | 140-0x02 |

**Contents** **page**

## 1 PIN ASSIGNMENTS

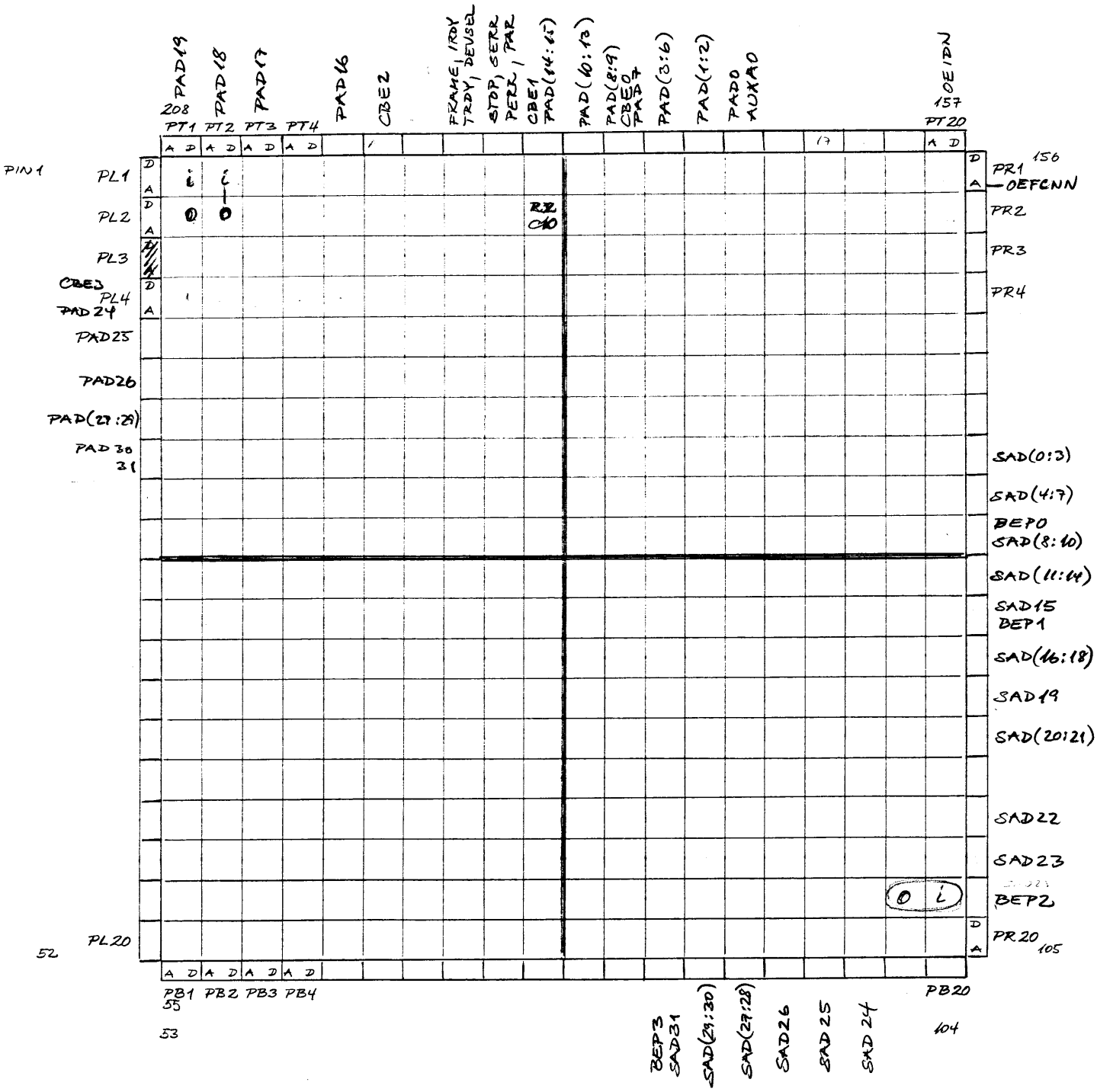
| Pin | Pad | Signal | Input PLC | Output PLC |
|---|---|---|---|---|
| 1 | VSS | | | |
| 2 | VSS | | | |
| 3 | PL1D | | | |
| 4 | PL2D | | | |
| 5 | PL4D | CBE3 | I/O-VDD5 | |
| 6 | A | PAD24 | | |
| 7 | PL5A | 25 | | |
| 8 | PL6A | 26 | | |
| 9 | PL7D | 27 | | |
| 10 | B | 28 | | |
| 11 | A | 29 | | |
| 12 | VDD | | | |
| 13 | PL8D | PAD30 | | |
| 14 | C | 31 | | |
| 15 | B | | | |
| 16 | A | | | |
| 17 | PL9D | INTAN | | |
| 18 | C | BN | | |
| 19 | B | CN | | |
| 20 | A | DN | | |
| 21 | VSS | | | |
| 22 | PL10D | CLK | | |
| 23 | C | | | |
| 24 | B | | | |
| 25 | A | | | |
| 26 | VDD | | | |
| 27 | PL11D | REQ1N | | |
| 28 | C | 2N | | |
| 29 | B | 3N | | |
| 30 | A | 4N | | |
| 31 | VSS | | | |
| 32 | PL12D | GNT1N | | |
| 33 | C | 2N | | |
| 34 | B | 3N | | |
| 35 | A | 4N | | |
| 36 | PL13D | | | |
| 37 | C | | | |
| 38 | B | | | |
| 39 | A | | | |
| 40 | VDD | | | |
| 41 | PL14D | | | |
| 42 | B | | | |
| 43 | PL15D | | | |
| 44 | B | | | |
| 45 | PL16D | | | |
| 46 | PL17D | | | |
| 47 | PL18D | | | |
| 48 | PL19D | | | |
| 49 | A | | | |
| 50 | PL20A | | | |
| 51 | VSS | | | |
| 52 | CCLK | | | |

| Pin | Pad | Signal | Input PLC | Output PLC |
|---|---|---|---|---|
| 53 | VSS | | ..... | ..... |
| 54 | VSS | | ..... | ..... |
| 55 | PB1A | ..... | ..... | ..... |
| 56 | PB2A | ..... | ..... | ..... |
| 57 | D | ..... | ..... | ..... |
| 58 | PB3D | ..... | ..... | ..... |
| 59 | PB4D | ..... | ..... | ..... |
| 60 | PB5D | ..... | ..... | ..... |
| 61 | PB6B | ..... | ..... | ..... |
| 62 | D | GAIN | ..... | ..... |
| 63 | PB7B | GAON | ..... | ..... |
| 64 | D | IRQ | ..... | ..... |
| 65 | VDD | | ..... | ..... |
| 66 | PB8A | BGOUTN | ..... | ..... |
| 67 | B | BGINN | ..... | ..... |
| 68 | C | BUSY | ..... | ..... |
| 69 | D | BRQ | ..... | ..... |
| 70 | PB9A | WR | ..... | ..... |
| 71 | B | WRN | ..... | ..... |
| 72 | C | ACK | ..... | ..... |
| 73 | D | ACKN | ..... | ..... |
| 74 | VSS | | ..... | ..... |
| 75 | PB10A | DS | ..... | ..... |
| 76 | B | DSN | ..... | ..... |
| 77 | C | ASACK | ..... | ..... |
| 78 | D | ASACKN | ..... | ..... |
| 79 | VSS | | ..... | ..... |
| 80 | PB11A | ACN | ..... | ..... |
| 81 | B | AC | ..... | ..... |
| 82 | B | PAS | ..... | ..... |
| 83 | D | PASN | ..... | ..... |
| 84 | VSS | | ..... | ..... |
| 85 | PB12A | ERRLEDN | I/O-VDD5 | ..... |
| 86 | B | ..... | ..... | ..... |
| 87 | C | ENEDRUN | ..... | ..... |
| 88 | D | ENEREC | ..... | ..... |
| 89 | ~~PB13A~~ | ~~.....~~ | I/O-HDC | ..... |
| 90 | PB13B | ..... | ..... | ..... |
| 91 | C | BEP3 | ..... | ..... |
| 92 | D | SAD31 | ..... | ..... |
| 93 | VDD | | ..... | ..... |
| 94 | PB14A | SAD30 | I/O-LDC | ..... |
| 95 | D | 29 | ..... | ..... |
| 96 | PB15A | 28 | ..... | ..... |
| 97 | D | 27 | ..... | ..... |
| 98 | INIT ~~PB16A~~ | 26 | INIT | ..... |
| 99 | PB17A | 25 | ..... | ..... |
| 100 | PB18A | 24 | ..... | ..... |
| 101 | PB19D | ~~.....~~ | ..... | ..... |
| 102 | PB20D | ~~.....~~ | ..... | ..... |
| 103 | VSS | | ..... | ..... |
| 104 | DONE | | ..... | ..... |

| Pin | Pad | Signal | Input PLC | Output PLC | |
|---|---|---|---|---|---|
| 105 | VSS | | | | |
| 106 | RESET | RESETN | | | |
| 107 | PRGM | | | | |
| 108 | M0 | | | | |
| 109 | PR19A | BEP2 | BEP2 | | |
| 110 | PR18A | SAD23 | SAD23 | | |
| 111 | PR17A | 22 | 22 | | |
| 112 | M1 | | | | |
| 113 | PR15A | SAD21 | SAD21 | | |
| 114 | D | 20 | 20 | | |
| 115 | PR14A | 19 | 19 | | |
| 116 | VDD | | | | |
| 117 | M2 | | | | |
| 118 | PR13B | SAD18 | SAD18 | | |
| 119 | C | 17 | 17 | | |
| 120 | D | 16 | 16 | | |
| 121 | M3 | | | | |
| 122 | PR12B | | | | |
| 123 | C | BEP1 | | | |
| 124 | D | SAD15 | | | |
| 125 | VSS | | | | |
| 126 | PR11A | SAD14 | | | |
| 127 | B | 13 | | | |
| 128 | C | 12 | | | |
| 129 | D | 11 | | | |
| 130 | VDD | | | | |
| 131 | PR10A | SAD10 | | | |
| 132 | B | 9 | | | |
| 133 | C | 8 | | | |
| 134 | D | BEP0 | | | |
| 135 | VSS | | | | |
| 136 | PR9A | SAD7 | | | |
| 137 | B | 6 | | | |
| 138 | C | 5 | | | |
| 139 | D | 4 | | | |
| 140 | PR8A | SAD3 | | | |
| 141 | B | 2 | | | |
| 142 | C | 1 | | | |
| 143 | D | 0 | | | |
| 144 | VDD | | | | |
| 145 | PR7A | | I/O-CS0 V | INP | |
| 146 | B | AUXD7 | | | input only |
| 147 | PR6B | 6 | | | |
| 148 | D | 5 | | | |
| 149 | PR5A | 4 | RD | INP | |
| 150 | D | 3 | | | |
| 151 | PR4A | 2 | | | |
| 152 | PR3A | 1 | WR | INP | |
| 153 | PR2A | 0 | | | |
| 154 | PR1A | OEFCNN | | | direct outp |
| 155 | VSS | | | | |
| 156 | RDCFN | | | | |

3C15A

| Pin | Pad | Signal | Input PLC | Output PLC |
|---|---|---|---|---|
| 157 | VSS | | ..... | ..... |
| 158 | VSS | | ..... | ..... |
| 159 | PT20D | OEIDN | ..... | ..... |
| 160 | PT19A | AUXA4 | ..I/O-RDY. | ..... |
| 161 | PT17D | .....3 | ..... | ..... |
| 162 | A | .....2 | ..... | ..... |
| 163 | PT16D | .....1 | ..... | ..... |
| 164 | PT15D | .....0 | ..... | ..... |
| 165 | A | PAD0. | ..... | ..... |
| 166 | PT14D | ....1. | ..... | ..... |
| 167 | B | ....2. | ..... | ..... |
| 168 | VDD | | ..... | ..... |
| 169 | PT13D | PAD3. | ..... | ..... |
| 170 | C | ....4. | ..... | ..... |
| 171 | B | ....5. | ..... | ..... |
| 172 | A | ....6. | ..... | ..... |
| 173 | PT12D | ....7. | ..... | ..... |
| 174 | C | CBE0. | ..... | ..... |
| 175 | B | PAD8. | ..... | ..... |
| 176 | PT13A | ....9. | ..... | ..... |
| 177 | VSS | | ..... | ..... |
| 178 | PT11D | PAD10. | ..... | ..... |
| 179 | C | ....11. | ..... | ..... |
| 180 | B | ....12. | ..... | ..... |
| 181 | A | ....13. | ..... | ..... |
| 182 | VSS | | ..... | ..... |
| 183 | PT10D | PAD14 | ..... | ..... |
| 184 | C | ....15. | ..... | ..... |
| 185 | B | CBE1. | ..... | ..... |
| 186 | A | ..... | ..... | ..... |
| 187 | VSS | | ..... | ..... |
| 188 | PT9D √ | PAR. | ..... | ..... |
| 189 | C √ | PERRN | ..... | ..... |
| 190 | B √ | SERRN | ..... | ..... |
| → 191 | DIN A | STOPN | ..DIN... | ..... |
| 192 | PT8D | DEVSELN | ..... | ..... |
| 193 | C | TRDYN | ..... | ..... |
| 194 | B | IRDYN | ..... | ..... |
| 195 | A √ | FRAMEN | ..DOUT.. | ..... |
| 196 | VDD | | ..... | ..... |
| 197 | 5D PT7D √ | CBE2 | ..... | ..... |
| 198 | 5A A √ | PAD16 | ..... | ..... |
| 199 | PT6C √ | ....17 | ..... | ..... |
| 200 | A √ | ....18 | ..TDI... | ..... |
| 201 | PT5A √ | ....19 | ..... | ..... |
| 202 | PT4A √ | ....20 | ..TMS... | to one during config |
| 203 | PT3A √ | ....21 | ..... | ..... |
| 204 | PT2A √ | ....22 | ..... | ..... |
| 205 | PT1D √ | ....23 | ..... | ..... |
| 206 | A | ..... | ..TCK... | to one during config |
| 207 | VSS | | ..... | ..... |
| 208 | RDDAT | | ..... | ..... |


PAD19
208
PT1
PAD18
PT2
PAD17
PT3
PT4
PAD16
CBE2
FRAME, IRDY
TRDY, DEVSEL
STOP, SERR
PERR, PAR
CBE1
PAD(14:15)
PAD(10:13)
PAD(8:9)
CBE0
PAD7
PAD(3:6)
PAD(1:2)
PAD0
AUXA0
OE IDN
157
PT20
156
PR1
OEFCNN
PR2
PR3
PR4
PIN 1
PL1
PL2
PL3
CBE3
PL4
PAD24
PAD25
PAD26
PAD(27:28)
PAD 30
31
SAD(0:3)
SAD(4:7)
BEP0
SAD(8:10)
SAD(11:14)
SAD15
BEP1
SAD(16:18)
SAD19
SAD(20:21)
SAD22
SAD23
BEP2
PR20
105
PL20
52
PB1
55
53
PB2
PB3
PB4
PB20
104
BEP3
SAD31
SAD(29:30)
SAD(27:28)
SAD26
SAD25
SAD24

POS = Rx Cy

# DESIGN NOTE

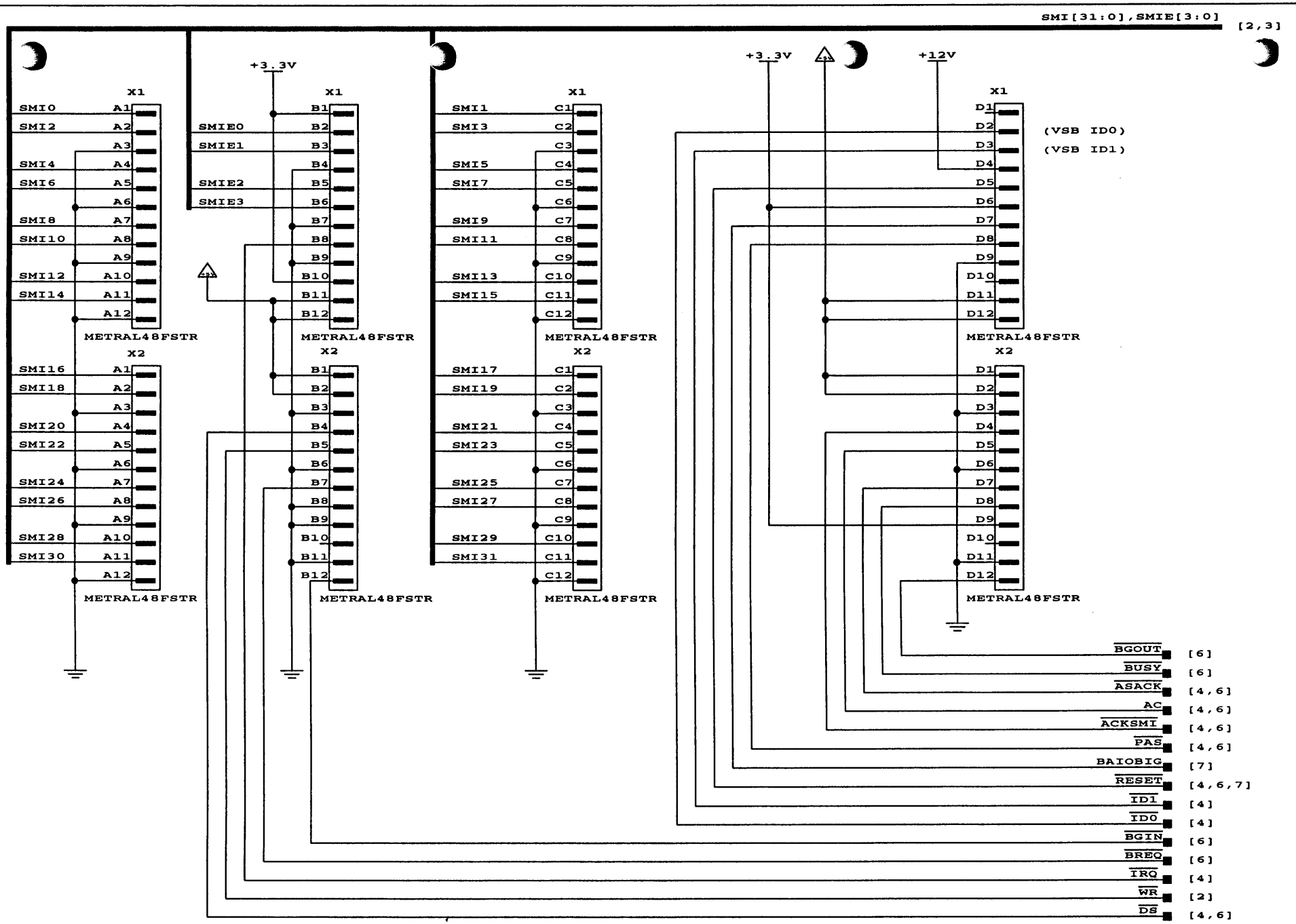
| | |
|---|---|
| Name: | BSP301 Pin Assignment |
| Date: | 1997-02-25 |
| Author: | aaj |
| Version: | 1.0 |
| Document ID: | |

**Contents** **page**

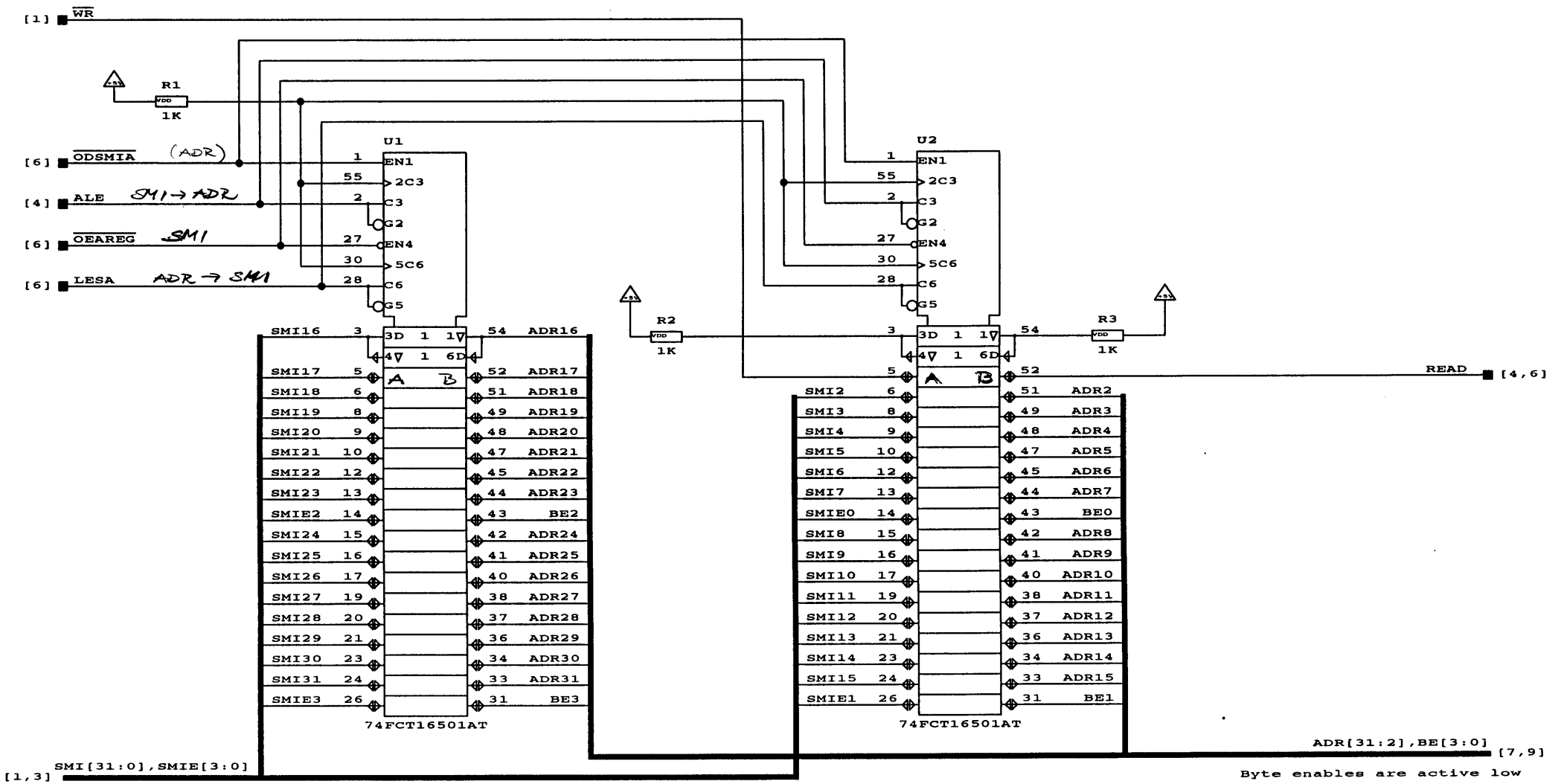
TPE 101

```
ACK   = Y4 & DS                                   ;   com
ENRD  = DS & READ & (Y6 + Y4)                     ;   com
REQI  = Y3 + Y5 ;        internal request         ;   com

ASACK = !Y1 & Y8 + Y2 + Y3 + Y5 + Y6 + Y4 & PAS   ;   com

ALE   = !ASACK & RDYI                             ;   com
```

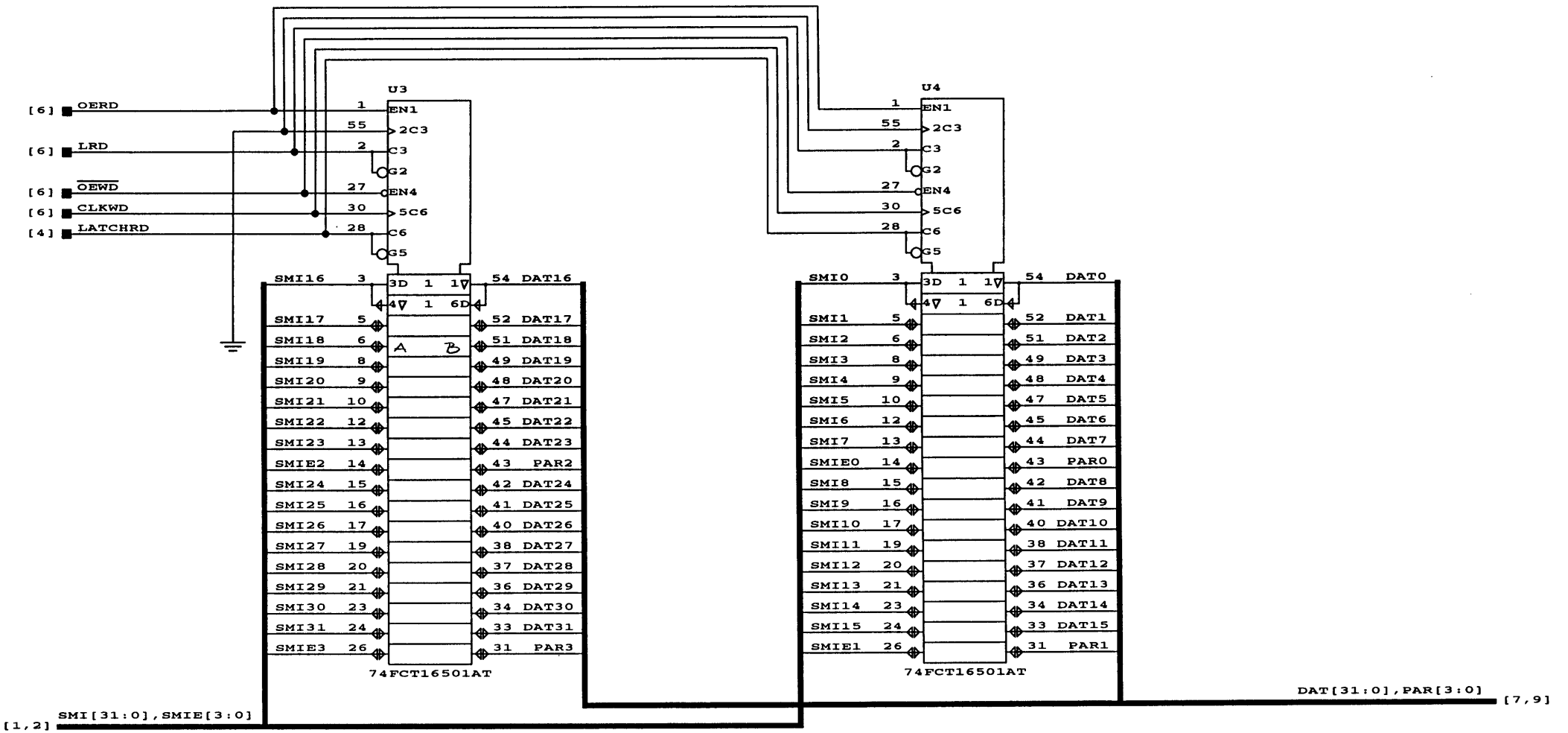
RDYI clears asynchront at reqi



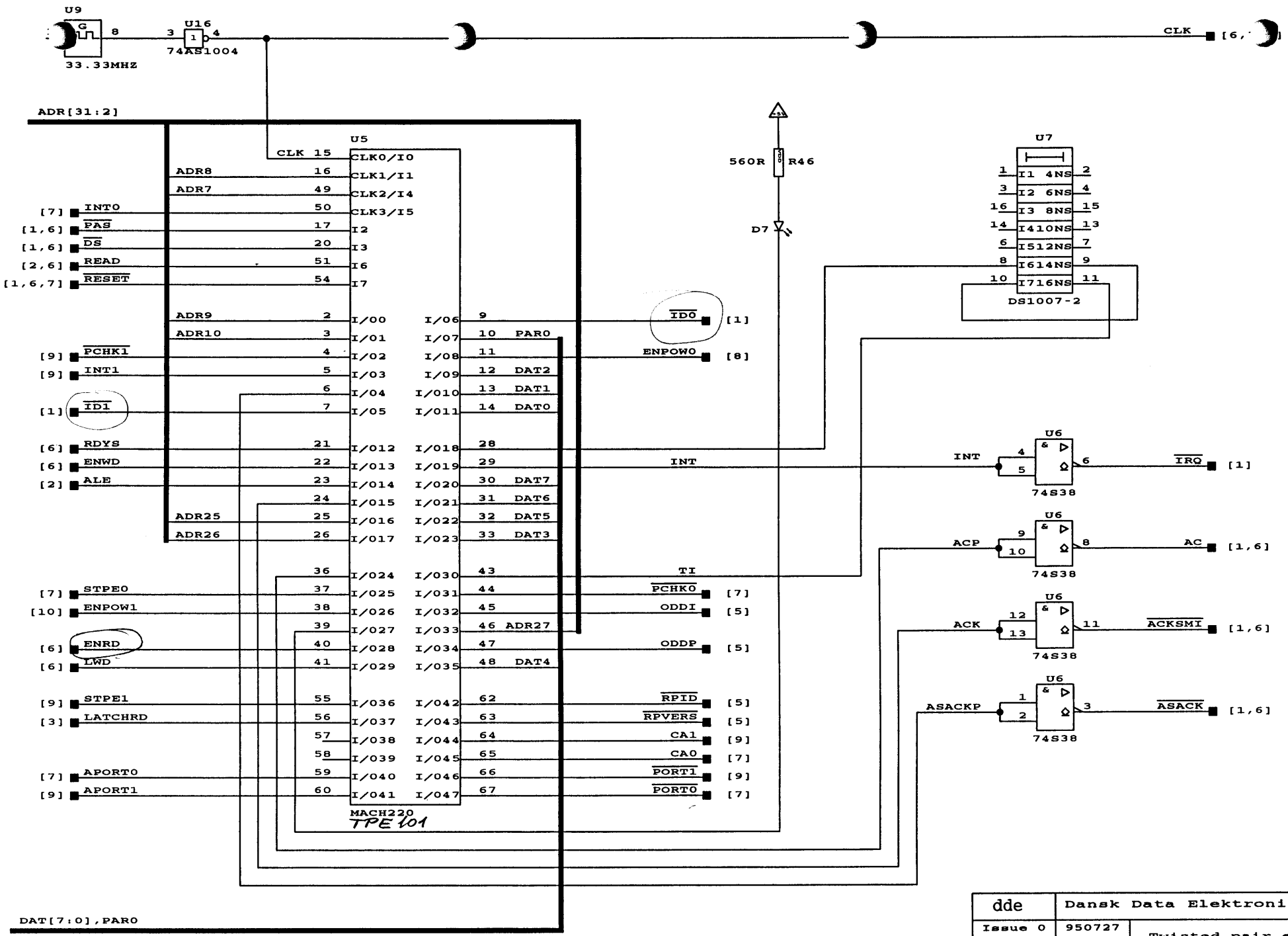

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 950727 | Twisted pair eth TPE301 SMI interface | |
| Issue 1 | 960103 | | |
| Issue 2 | | | |
| Issue 3 | | File: tpe | Page: 1 of 11 |



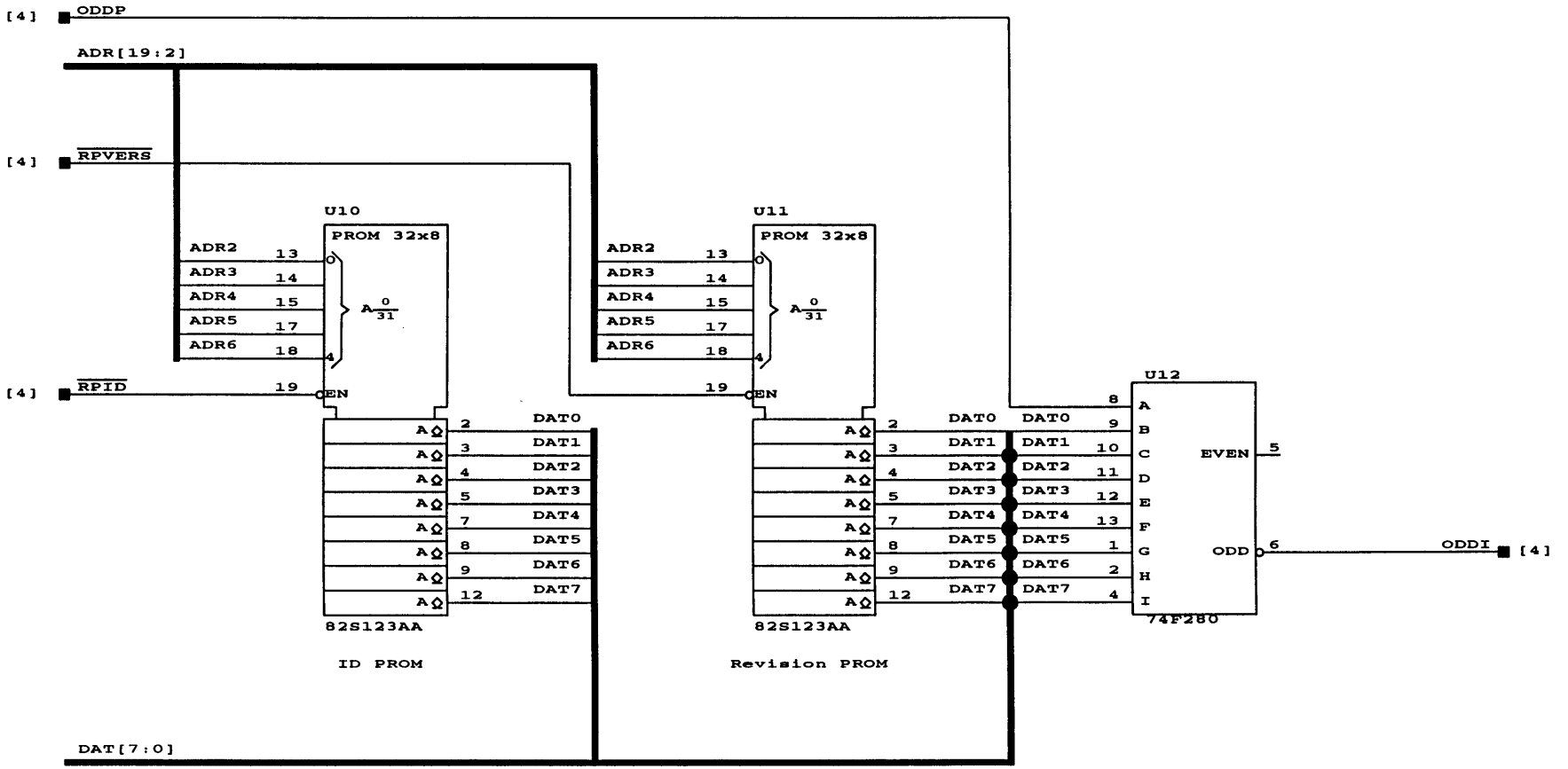

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 950727 | Twisted pair eth TPE301 AREG | |
| Issue 1 | 960103 | | |
| Issue 2 | | | |
| Issue 3 | | File: tpe | Page:2 of 11 |



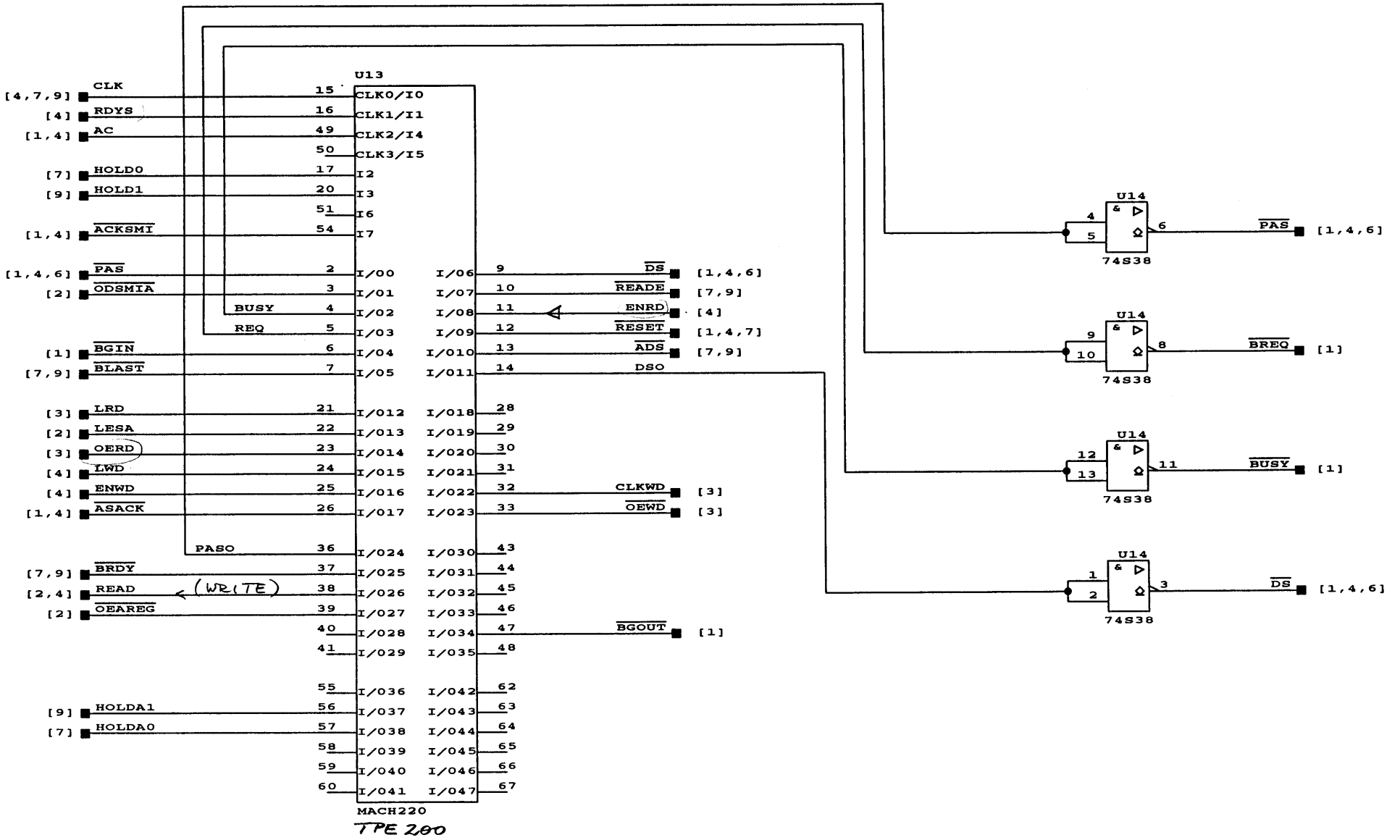

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 950727 | Twisted pair eth TPE301 DREG | |
| Issue 1 | 960103 | | |
| Issue 2 | | | |
| Issue 3 | | File: tpe | Page:3 of 11 |



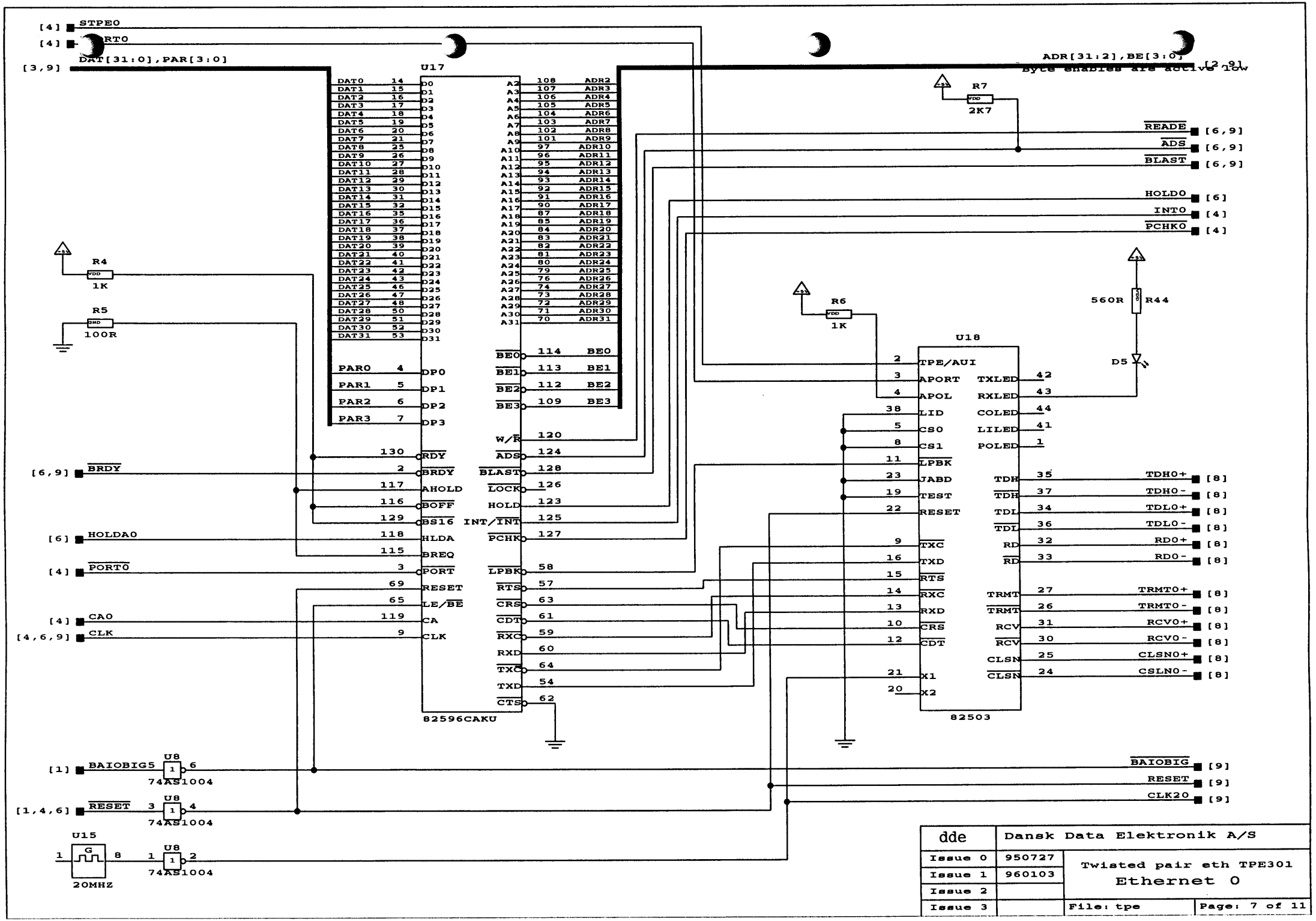

| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 950727 | Twisted pair eth TPE301 SMI control |
| Issue 1 | 960103 | |
| Issue 2 | | |
| Issue 3 | | File: tpe Page: 4 of 11 |

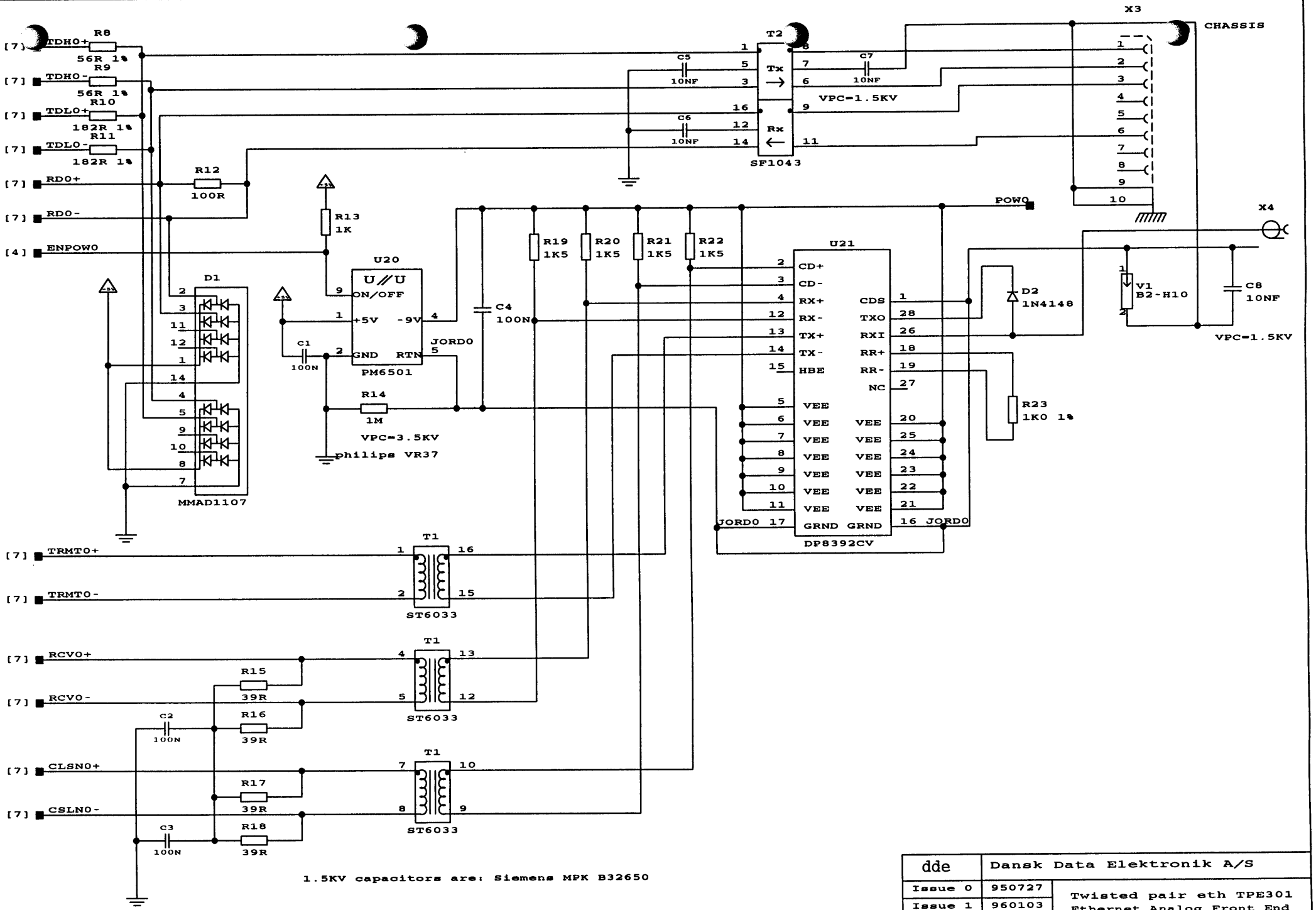



| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 950727 | Twisted pair eth TPE301 ID and Revision PROMs | |
| Issue 1 | 960103 | | |
| Issue 2 | | | |
| Issue 3 | | File: tpe | Page: 5 of 11 |

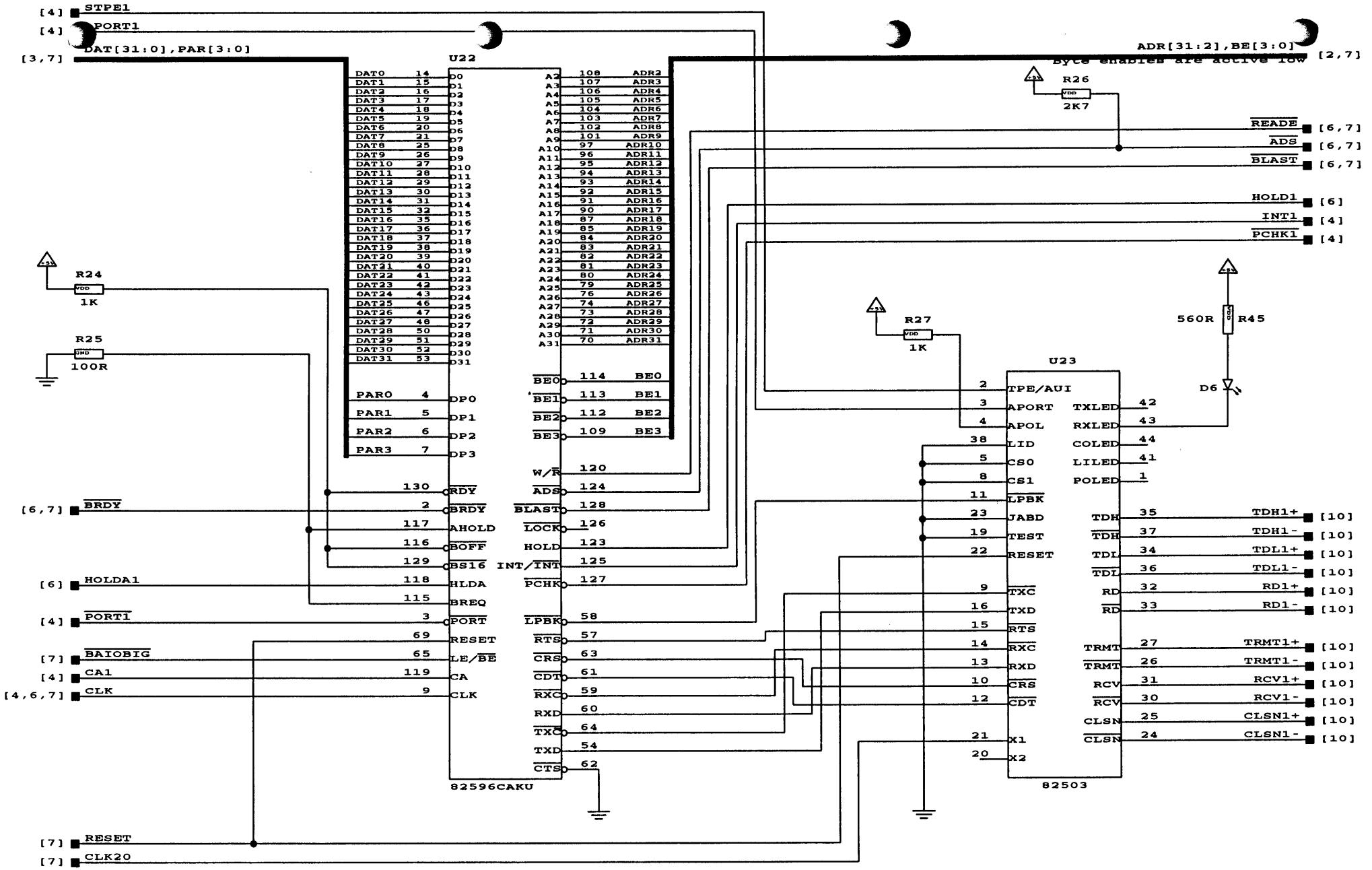



| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 950727 | Twisted pair eth TPE301 Ethernet Control | |
| Issue 1 | 960103 | | |
| Issue 2 | | | |
| Issue 3 | | File: tpe | Page: 6 of 11 |

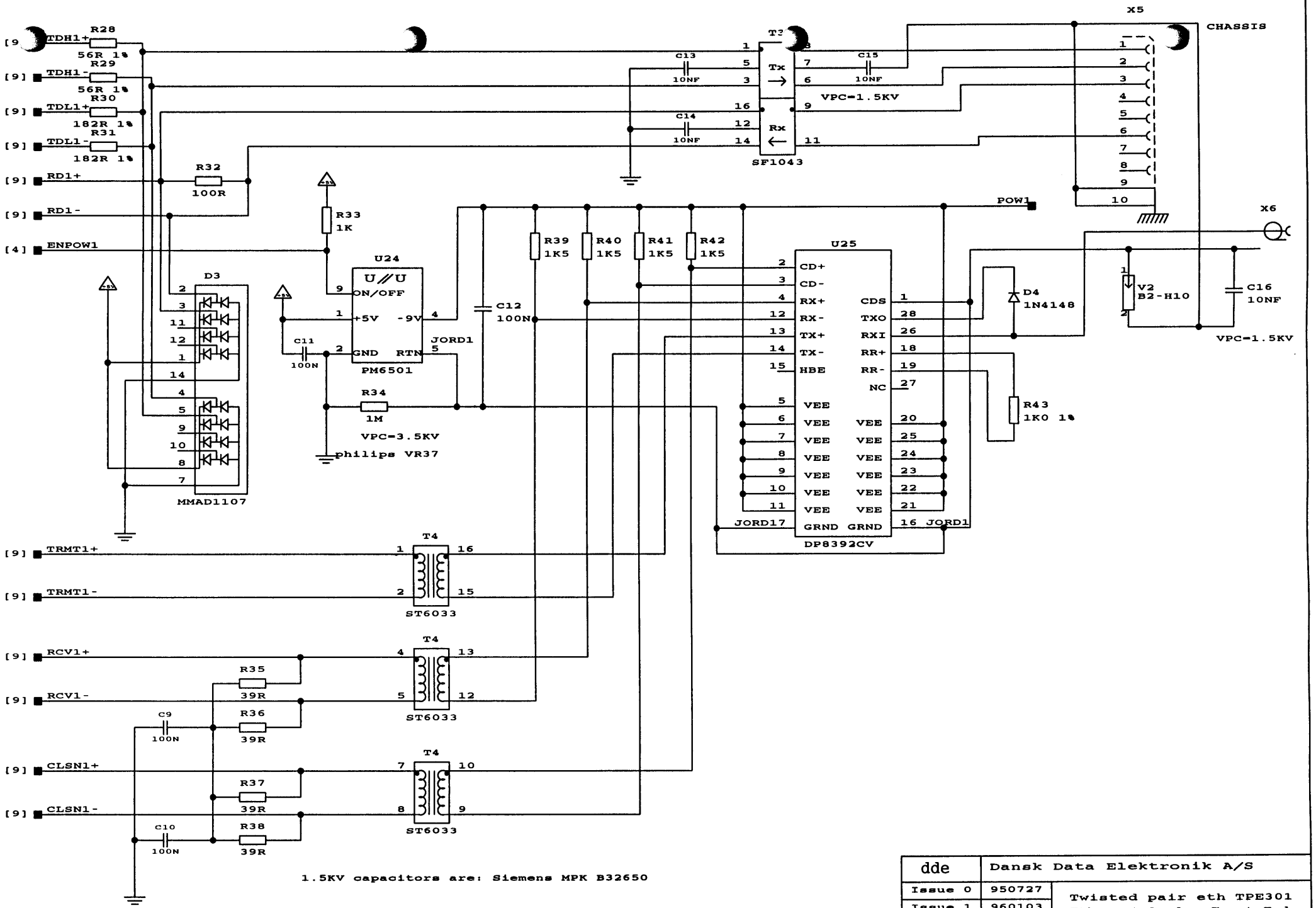

[4] STPE0
[4] PORT0
[3,9] DAT[31:0],PAR[3:0]
ADR[31:2],BE[3:0] [2,9]
Byte enables are active low
U17
DAT0 14 D0
DAT1 15 D1
DAT2 16 D2
DAT3 17 D3
DAT4 18 D4
DAT5 19 D5
DAT6 20 D6
DAT7 21 D7
DAT8 25 D8
DAT9 26 D9
DAT10 27 D10
DAT11 28 D11
DAT12 29 D12
DAT13 30 D13
DAT14 31 D14
DAT15 32 D15
DAT16 35 D16
DAT17 36 D17
DAT18 37 D18
DAT19 38 D19
DAT20 39 D20
DAT21 40 D21
DAT22 41 D22
DAT23 42 D23
DAT24 43 D24
DAT25 46 D25
DAT26 47 D26
DAT27 48 D27
DAT28 50 D28
DAT29 51 D29
DAT30 52 D30
DAT31 53 D31
PAR0 4 DP0
PAR1 5 DP1
PAR2 6 DP2
PAR3 7 DP3
A2 108 ADR2
A3 107 ADR3
A4 106 ADR4
A5 105 ADR5
A6 104 ADR6
A7 103 ADR7
A8 102 ADR8
A9 101 ADR9
A10 97 ADR10
A11 96 ADR11
A12 95 ADR12
A13 94 ADR13
A14 93 ADR14
A15 92 ADR15
A16 91 ADR16
A17 90 ADR17
A18 87 ADR18
A19 85 ADR19
A20 84 ADR20
A21 83 ADR21
A22 82 ADR22
A23 81 ADR23
A24 80 ADR24
A25 79 ADR25
A26 76 ADR26
A27 74 ADR27
A28 73 ADR28
A29 72 ADR29
A30 71 ADR30
A31 70 ADR31
BE0 114 BE0
BE1 113 BE1
BE2 112 BE2
BE3 109 BE3
W/R 120
ADS 124
BLAST 128
LOCK 126
HOLD 123
INT/INT 125
PCHK 127
LPBK 58
RTS 57
CRS 63
CDT 61
RXC 59
RXD 60
TXC 64
TXD 54
CTS 62
82596CAKU
R4 1K
R5 100R
R7 2K7
R6 1K
R44 560R
D5
READE [6,9]
ADS [6,9]
BLAST [6,9]
HOLD0 [6]
INT0 [4]
PCHK0 [4]
U18
2 TPE/AUI
3 APORT
4 APOL
38 LID
5 CS0
8 CS1
11 LPBK
23 JABD
19 TEST
22 RESET
9 TXC
16 TXD
15 RTS
14 RXC
13 RXD
10 CRS
12 CDT
21 X1
20 X2
TXLED 42
RXLED 43
COLED 44
LILED 41
POLED 1
TDH 35 TDH0+ [8]
TDH 37 TDH0- [8]
TDL 34 TDL0+ [8]
TDL 36 TDL0- [8]
RD 32 RD0+ [8]
RD 33 RD0- [8]
TRMT 27 TRMT0+ [8]
TRMT 26 TRMT0- [8]
RCV 31 RCV0+ [8]
RCV 30 RCV0- [8]
CLSN 25 CLSN0+ [8]
CLSN 24 CSLN0- [8]
82503
[6,9] BRDY
[6] HOLDA0
[4] PORT0
[4] CA0
[4,6,9] CLK
130 RDY
2 BRDY
117 AHOLD
116 BOFF
129 BS16
118 HLDA
115 BREQ
3 PORT
69 RESET
65 LE/BE
119 CA
9 CLK
[1] BAIOBIG5
[1,4,6] RESET
U15
20MHZ
U8
74AS1004
BAIOBIG [9]
RESET [9]
CLK20 [9]
dde
Dansk Data Elektronik A/S
Issue 0 950727
Issue 1 960103
Issue 2
Issue 3
Twisted pair eth TPE301
Ethernet 0
File: tpe
Page: 7 of 11



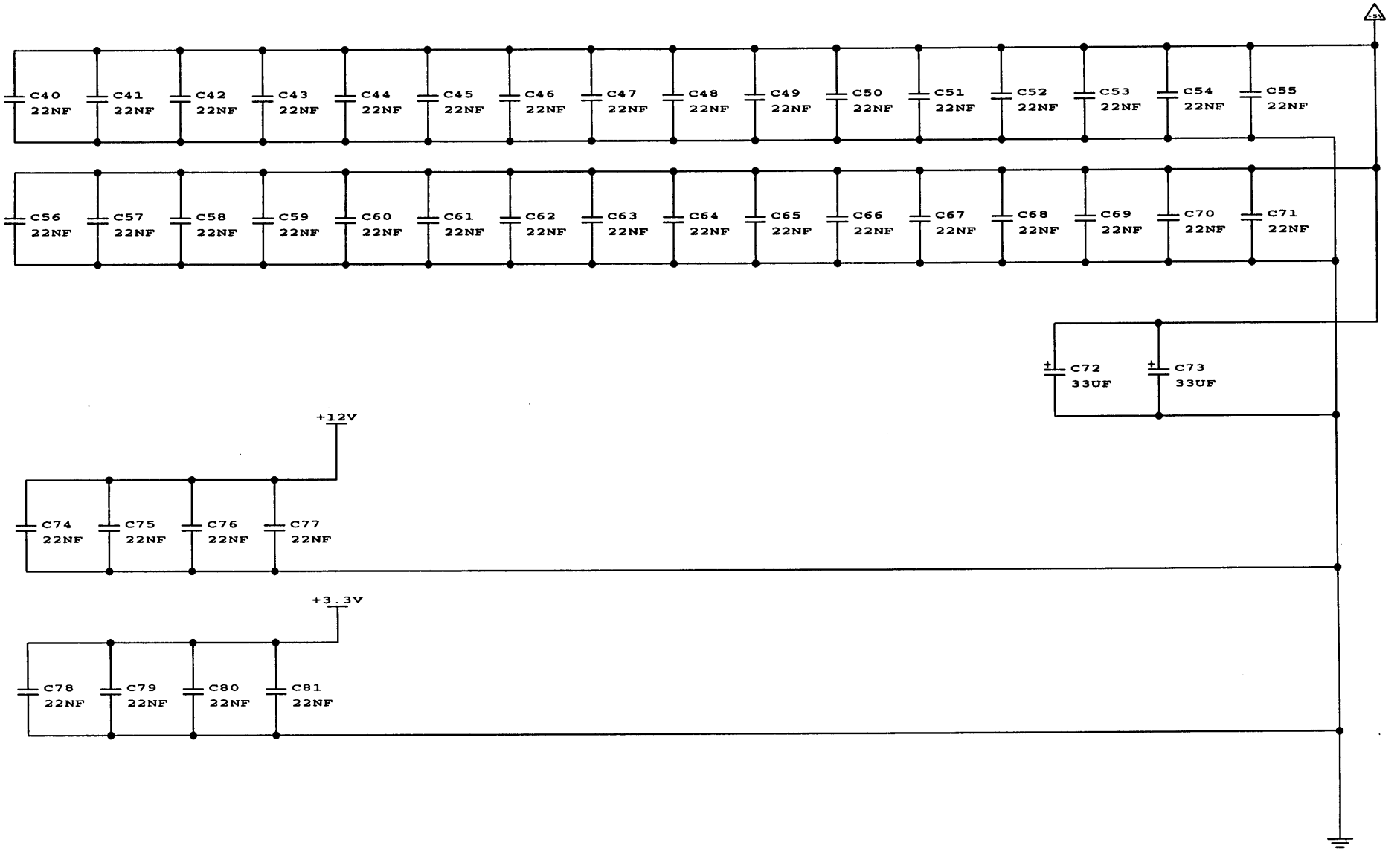

| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 950727 | Twisted pair eth TPE301 Ethernet Analog Front End | |
| Issue 1 | 960103 | | |
| Issue 2 | | | |
| Issue 3 | | File: tpe | Page:8 of 11 |




| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 950727 | Twisted pair eth TPE301 Ethernet 1 |
| Issue 1 | 960103 | |
| Issue 2 | | |
| Issue 3 | | File: tpe Page: 9 of 11 |




| dde | Dansk Data Elektronik A/S | |
|---|---|---|
| Issue 0 | 950727 | Twisted pair eth TPE301 Ethernet Analog Front End |
| Issue 1 | 960103 | |
| Issue 2 | | |
| Issue 3 | | File: tpe Page:10 of 11 |

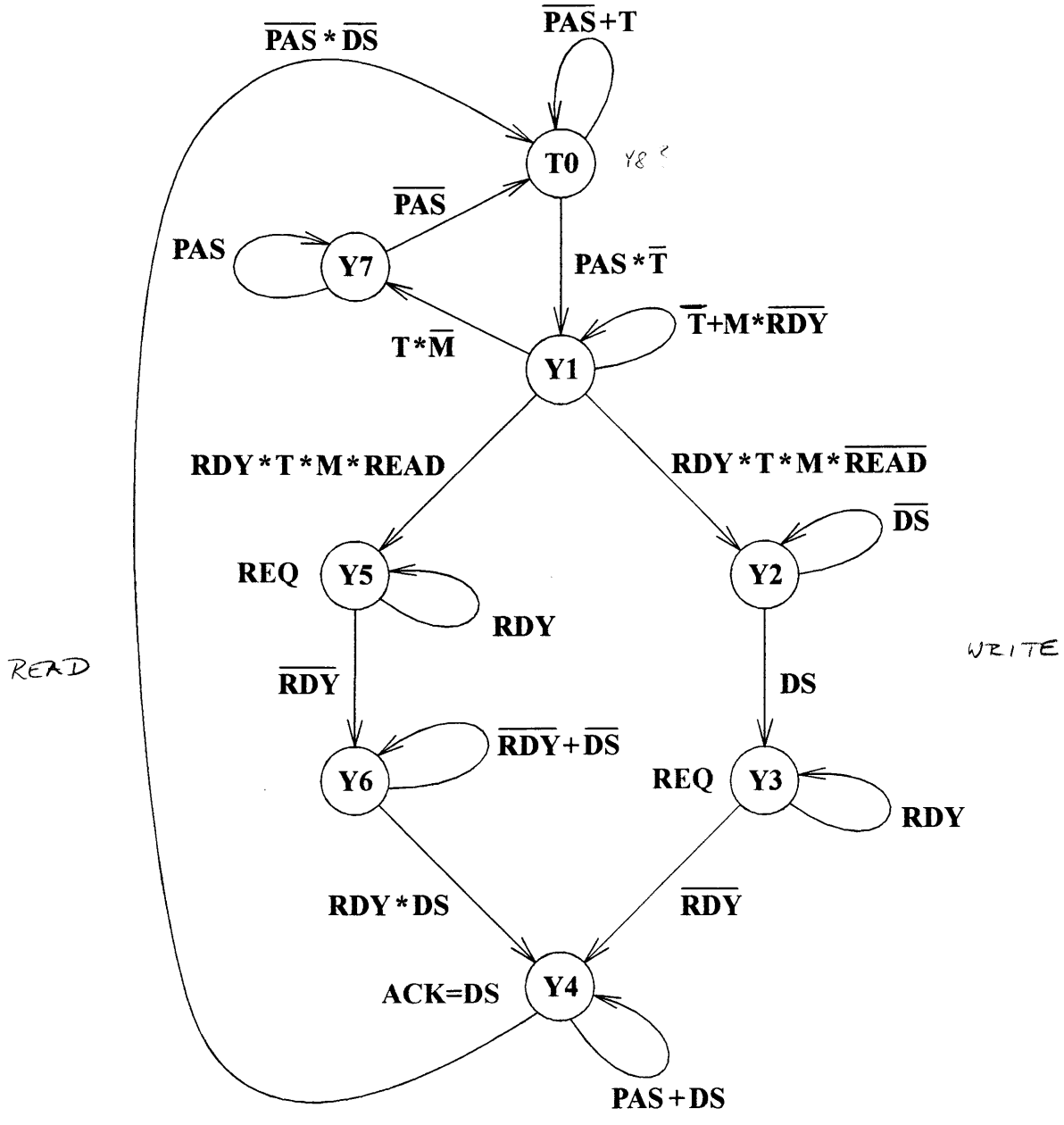



| dde | Dansk Data Elektronik A/S | | |
|---|---|---|---|
| Issue 0 | 950727 | Twisted pair eth TPE301<br>Decoupling capacitors | |
| Issue 1 | 960103 | | |
| Issue 2 | | | |
| Issue 3 | | File: tpe | Page:11 of 11 |

# TPE301 SMICONT-Tilstandsdiagram

Dato: 95.10.20

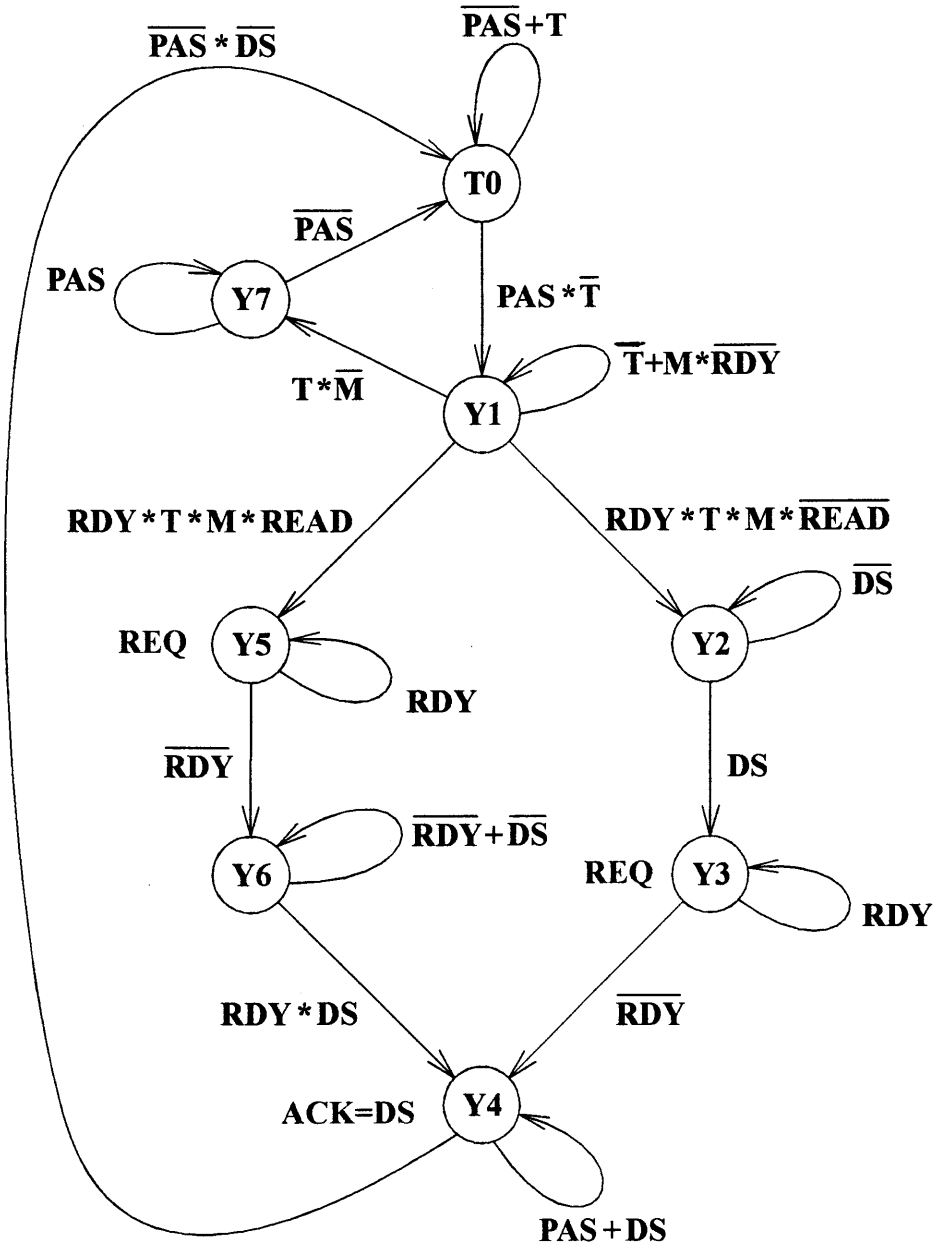
KAN



# TPE301 SMICONT-Tilstandsdiagram

Dato: 95.10.20

KAN



```
"control PAL for 4 channel ethernet submodule
"                                                    kan 26/5-1994
module  tpe200;

"Channel 2 and 3 removed.kan 21/8-1995
"Changed                                             kan 3/11-1995


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

tpe200 device 'mach220a'; "speed 15 ns
x, z, c = .X., .Z., .C.;

"#########################################################################
"                        Signal description
"#########################################################################

clk     pin  15 ;"clock input pin, 33 Mhz.

!res    pin  12 ;"reset input pin.

"#########################################################################
"              SMI signals used for arbitration
"#########################################################################

req     pin   5 istype'reg_d,buffer';"bus request output to SMI
busy    pin   4 istype'reg_t,buffer';"busy output to SMI

!bgin    pin   6 ;                   "bus grant input from SMI
 bgout   pin  47 istype'com,invert';"bus grant output to next module

!pasi    pin   2 ;                    "physical address strobe.
                                     "input signal from SMI
"#########################################################################
"     Signals to/from ethernet controllers used for arbitration
"#########################################################################

"controller 0

hold0   pin  17 ;"hold request signal
holda0  pin  57 istype'reg_d,buffer';"hold acknowledge


"controller 1

hold1   pin  20 ;"hold request signal
holda1  pin  56 istype'reg_d,buffer';"hold acknowledge

odsmia  pin   3 istype'reg_t,invert';"output disable address from SMI

rdys    pin  16 ;"ready input signal from SMISCU. zero during buffered
                                                  "write.
"#########################################################################
"                  nodes used for arbitration
"#########################################################################
"state variables used in state machine close to SMI

n0      node    istype'reg_d,buffer';"also used to latch bgin low -> Z
n1      node    istype'reg_d,buffer';"also used to latch bgin low -> y
n2      node    istype'reg_d,buffer';

Z       node    istype'com,buffer';"bgin low when n0 set
y       node    istype'com,buffer';"bgin low when n1 set
gah     node    istype'reg_d,buffer';"go ahead signal to state machines
                                     "close to  controllers

"state variables used in state machines close to controllers

n0a     node    istype'reg_d,buffer';"state variable for controller 0
n0b     node    istype'reg_d,buffer';"state variable for controller 0

t       node    istype'reg_t,buffer';"time out signal for holda

"#########################################################################
"                  definition of states
"#########################################################################
"states used in state machine close to SMI

s0      = ^b000;"idle state
s1      = ^b100;"request from ethernet; wait for bgin
s2      = ^b010;"wait for bgin deactivated
s3      = ^b001;"access requested from other submodule; generate bgout

"input to state machine:
ina     = [bgin, req.q, gah.q, y, Z];

"states used in state machines close to controllers

t0      = ^b00;" idle state; controller 0 has priority
t1      = ^b01;" in cycle: controller 0
t2      = ^b10;" idle state; controller 1 has priority
t3      = ^b11;" in cycle: controller 1

"input to state machine:
in0     = [hold0, hold1, gah.q];

"#########################################################################
"                   clock and reset equations
"#########################################################################
equations;
req.c           = clk;
busy.c          = clk;
```

```
req.ar          = res;
busy.ar         = res;

holda0.c        = clk;
holda1.c        = clk;

holda0.ar       = res;
holda1.ar       = res;

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

n0.ar           = res;
n1.ar           = res;
n2.ar           = res;

gah.c           = clk;

n0a.c           = clk;
n0b.c           = clk;

gah.ar          = res;

n0a.ar          = res;
n0b.ar          = res;

t.c             = clk;
t.ar            = res;

"#########################################################################
"                equations for latched variables
"#########################################################################

y = !bgin & n1.q # y & n1.q;

Z = !bgin & n0.q # Z & n0.q;

!bgout = bgin & n0.q & !Z;"Generate bgout in state s3 until !bgin

"#########################################################################
"                State machine close to SMI
"#########################################################################
state_diagram [n2, n1, n0];

state s0:
"idle state

"            bgin, req.q, gah.q, y, Z
case(ina == [ 0,    0,     x,     x, x]): s0; "wait for something
    (ina == [ x,    1,     x,     x, x]): s1; "access from ethernet
    (ina == [ 1,    0,     x,     x, x]): s3; "access from other module
endcase;

state s1:
"access from ethernet module. wait for gah

gah.d = bgin & !pasi & rdys;

busy.t = !busy.q & gah.q;

odsmia.t = !odsmia.q & gah.q; "disable address from SMI

"            bgin, req.q, gah.q, y, Z
case(ina == [ x,    x,     0,     x, x]): s1; "wait for gah
    (ina == [ x,    x,     1,     x, x]): s2;
endcase;

state s2:
"access from ethernet module. wait for y ( !bgin )


"            bgin, req.q, gah.q, y, Z
case(ina == [ x,    x,     x,     0, x]): s2; "wait for y
    (ina == [ x,    x,     x,     1, x]): s0; "
endcase;


state s3:
"access from other submodule to BAIO. wait for Z ( !bgin )
"generate bgout until Z ( !bgin). see equations.

"            bgin, req.q, gah.q, y, Z
case(ina == [ x,    x,     x,     x, 0]): s3; "wait for Z
    (ina == [ x,    x,     x,     x, 1]): s0; "
endcase;

"#########################################################################
"                State machine for controller 0 and 1
"#########################################################################

state_diagram [n0b, n0a];

state t0:
"idle state. controller 0 has priority

req.d   = hold0 # hold1;

t.t      =  t.q & (hold0 # hold1)& gah.q;"clear timer.
holda0.d =  hold0          & gah.q;"set hold_acknowledge 0
holda1.d = !hold0 &  hold1 & gah.q;"set hold_acknowledge 1
"            h0 h1 ga
case(in0 == [x, x, 0]): t0;"idle controller 0 has priority
    (in0 == [1, x, 1]): t1;"in cycle controller 0
```

```
    (in0 == [0, 1, 1]): t3;"in cycle controller 1
endcase;

state t1:
"in cycle. Controller 0. Wait for !hold0;

holda0.d =        hold0 & !t.q; "hold hold_acknowledge until t
busy.t =  busy.q & !hold0;      "clear busy after hold0

odsmia.t =  odsmia.q & !hold0; "enable address from SMI after hold0

"             h0 h1 ga
case(in0 == [1, x, x]): t1;"in cycle controller 0. Wait for !hold0
    (in0 == [0, x, x]): t2;
endcase;

state t2:
"idle state. controller 1 has priority

req.d   = hold0 # hold1;

t.t      =  t.q & (hold0 # hold1)& gah.q;"clear timer.
holda0.d =  hold0 & !hold1 & gah.q;"set hold_acknowledge 0
holda1.d =          hold1 & gah.q;"set hold_acknowledge 1

"             h0 h1 ga
case(in0 == [x, x, 0]): t2;"idle controller 0 has priority
    (in0 == [x, 1, 1]): t3;"in cycle controller 1
    (in0 == [1, 0, 1]): t1;"in cycle controller 0
endcase;

state t3:
"in cycle. Controller 1. Wait for !hold1;

holda1.d =        hold1 & !t.q; "hold hold_acknowledge until t
busy.t =  busy.q & !hold1;      "clear busy after hold1

odsmia.t =  odsmia.q & !hold1; "enable address from SMI after hold0

"             h0 h1 ga
case(in0 == [x, 1, x]): t3;"in cycle controller 1. Wait for !hold1
    (in0 == [x, 0, x]): t0;
endcase;

"#########################################################################
"            Definition of pins for address and data phase
"#########################################################################

declarations;

"Input signals from ethernet controllers

!ads    pin  13 ;"address strobe from ethernet controllers
!reade  pin  10 ;"read signal from ethernet controllers
!blast  pin   7 ;"last data transfer in cycle

"Output signals to ethernet controllers.

brdy    pin  37 istype'reg_d,invert';"burst ready signal

"Output signals to module

oeareg  pin  39 istype'com,invert';"output enable address register
lesa    pin  22 istype'com,buffer';"open address latch. slave address
oewd    pin  33 istype'com,invert';"output enable write data to SMI
oerd    pin  23 istype'com,buffer';"output enable read data from SMI
lrd     pin  21 istype'com,buffer';"open latch read data
clkwd   pin  32 istype'com,buffer';"clock write data

"Input signals from SMI

!asack  pin  26 ;"address acknowledge to synchronous part
!ac     pin  49 ;"address complete to asynchronous part. polarity 951018
!acki   pin  54 ;"data handshake signal to asynchronous part
!dsi    pin   9 ;"data strobe input from SMI. to generate oewd

"Output signals to SMI

paso    pin  36 istype'reg_t,buffer';"Physical address strobe
ds      pin  14 istype'com,buffer'  ;"data handshake signal
write   pin  38 istype'com,invert'  ;"read/write signal to SMI

"Input signals from SMISCU

enrd    pin  11 ;"output enable data to SMI. enrd & dsi added to oewd
enwd    pin  25 ;"output enable data from SMI. enwd added to oerd
lwd     pin  24 ;"latch data from SMI. added to lrd.
"Nodes in synchronous part

dro     node    istype'reg_t,buffer';"handshake signal to async. part
enr     node    istype'reg_d,buffer';"enable signal to async. read part
enw     node    istype'reg_d,buffer';"enable signal to async. write part

asacks  node    istype'reg_d,buffer';"synchronized asack from SMI
fini    node    istype'reg_d,buffer';"synchronized end signal to state machine

cycle   node    istype'reg_d,buffer';"cycle in progress
n3      node    istype'reg_d,buffer';"state variable
n4      node    istype'reg_d,buffer';"state variable

ackd1   node    istype'com,buffer';"state variable
ackd2   node    istype'com,buffer';"state variable
ackd3   node    istype'com,buffer';"state variable
ack     node    istype'com,buffer';"state variable
```

```
"#########################################################################
"                    nodes in asynchronous part
"#########################################################################

"state variables for read cycles
y1      node    istype'com,buffer';
y2      node    istype'com,buffer';
y3      node    istype'com,buffer';
y4      node    istype'com,buffer';
y5      node    istype'com,buffer';
y6      node    istype'com,buffer';
finale  node    istype'com,buffer'; "blast # !enr.q latched by ds
                                    "latch open when ds = 1.
tom     node    istype'com,buffer';


"state variables for write cycles
u1      node    istype'com,buffer';
u2      node    istype'com,buffer';
u3      node    istype'com,buffer';
u4      node    istype'com,buffer';
u5      node    istype'com,buffer';
u6      node    istype'com,buffer';

dri     node    istype'com,buffer';

"Definition af states used in synchronous part

st0     = ^b00;"idle state
st1     = ^b01;"address phase
st2     = ^b11;"data phase
st3     = ^b10;"end data phase. wait for posted write

"#########################################################################
"                    Clock equations
"#########################################################################

equations;
paso.c  = clk;

paso.ar = res;

dro.c   = clk;
enr.c   = clk;
enw.c   = clk;

dro.ar  = res;
enr.ar  = res;
enw.ar  = res;

odsmia.c  = clk;

odsmia.ar = res;

asacks.c= clk;
fini.c  = clk;

asacks.ar = res;
fini.ar   = res;

brdy.c  = clk;
brdy.ar = res;

cycle.c = clk;
n3.c    = clk;
n4.c    = clk;

cycle.ar = res;
n3.ar    = res;
n4.ar    = res;

write.oe = cycle.q;

asacks.d = asack;

cycle.d = !res & (!cycle.q & ads # cycle.q & (!brdy.q # !blast));

state_diagram [n4, n3];

state st0:
"idle state. Wait for cycle.

paso.t = !paso.q & cycle.q & !asacks.q;
!oeareg = cycle.q & !asacks.q;
lesa    = cycle.q;
!write  = !reade & cycle.q;

dro.t  = cycle.q & !asacks.q & ( reade &  dro.q # !reade & !dro.q);

"read  cycles start with dro = 0;
"write cycles start with dro = 1;

case([cycle.q,asacks.q]  == [x,1]): st0; "wait
    ([cycle.q,asacks.q]  == [0,x]): st0; "wait
    ([cycle.q,asacks.q]  == [1,0]): st1; "start address phase
endcase;

state st1:
"address phase. Wait for asacks

!oeareg = !asacks.q;
!write  = !reade & cycle.q;
enr.d   =  reade & asacks.q;"enable asynchronous read
enw.d   = !reade & asacks.q;"enable asynchronous write
```

```
case(asacks.q == 0): st1; "wait for asacks
    (asacks.q == 1): st2; "start data phase
endcase;

state st2:
"data phase. Wait for last data

!write  = !reade & cycle.q # enw.q;
enr.d  =  enr.q & (!brdy.q # !blast);"enable async. read
enw.d  =  enw.q;                     "enable asynchronous write

brdy.d = !brdy.q & ( dro.q & !dri # !dro.q &  dri);

dro.t = brdy.q;

"dro.t = !enw.q & brdy.q & !blast # enw.q & brdy.q;

case([brdy.q,blast] == [0,x] ): st2; "wait ready
    ([brdy.q,blast] == [x,0] ): st2; "wait for blast
    ([brdy.q,blast] == [1,1] ): st3; "last data
endcase;

state st3:
"end data phase. Wait for posted write

fini.d  =  enw.q & (!dro.q &  dri #  dro.q & !dri)
        # !enw.q & !y4 & !y3 & !y2 & !y1;

!write  = enw.q;

enw.d  = enw.q & !fini.q; "clear enw
paso.t = paso.q & fini.q; "clear paso

case(fini.q == 0): st3; "wait
    (fini.q == 1): st0; "end
endcase;

"#########################################################################
"                  noise filter on ack
"#########################################################################

equations;

ackd1 = acki;
ackd2 = ackd1;
ackd3 = ackd2;
ack = acki&ackd3 # ack&ackd3 # ack&acki;

"#########################################################################
"                    asynchronous part
"#########################################################################

"ac er erstattet af blast

y1 =
   !res & !y6 & !y5 & !y4 & !y3 & !y2       & enr.q                "T0+Y1
 # !res & !y6 & !y5 &  y4 & !y3 & !y2       & !ack & !finale       "Y4+Y14
 # !res & !y6 & !y5       & !y3 & !y2 &  y1 & !ack & !finale       "Y1+Y14
 # !res & !y6 & !y5 & !y4 & !y3 & !y2 &  y1;                       "Y1

y2 =
   !res & !y6 & !y5 & !y4 & !y3       &  y1 &  ack & !dro.q       "Y1+Y12
 # !res & !y6 & !y5 & !y4 & !y3 &  y2       &  ack & !dro.q       "Y2+Y12
 # !res & !y6 & !y5 & !y4 & !y3 &  y2 & !y1;                      "Y2

y3 =
   !res & !y6 & !y5 & !y4       &  y2 & !y1 & !ack & !finale      "Y2+Y32
 # !res & !y6 & !y5 & !y4 &  y3       & !y1 & !ack & !finale      "Y3+Y32
 # !res & !y6 & !y5 & !y4 &  y3 & !y2 & !y1;                      "Y3

y4 =
   !res & !y6 & !y5       &  y3 & !y2 & !y1 &  ack &  dro.q      "Y3+Y34
 # !res & !y6 & !y5 &  y4       & !y2 & !y1 &  ack &  dro.q      "Y4+Y34
 # !res & !y6 & !y5 &  y4 & !y3 & !y2 & !y1;                          "Y4

y5 =
   !res & !y6       & !y4 & !y3 &  y2 & !y1 & !ack &  finale      "Y2+Y25
 # !res & !y6 &  y5 & !y4 & !y3       & !y1 & !fini.q;            "Y5+Y25

y6 =
   !res &       !y5 &  y4 & !y3 & !y2 & !y1 & !ack &  finale      "Y4+Y46
 # !res &  y6 & !y5       & !y3 & !y2 & !y1 & !fini.q;            "Y6+Y46


finale =  !ds & finale
        # ds & (blast # !enr.q)
        # finale & (blast # !enr.q);

tom = ac;

u1 =
   !res & !u6 & !u5 & !u4 & !u3 & !u2       &  enw.q               "T0+U1
 # !res &  u6 & !u5 & !u4 & !u3 & !u2       & !ack                 "U6+U16
 # !res &       !u5 & !u4 & !u3 & !u2 &  u1 & !ack & enw.q        "U1+U16
 # !res & !u6 & !u5 & !u4 & !u3 & !u2 &  u1 &  enw.q;             "U1

u2 =
   !res & !u6 & !u5 & !u4 & !u3       &  u1 & !dro.q & enw.q      "U1+U12
 # !res & !u6 & !u5 & !u4 & !u3 &  u2       & !dro.q & enw.q      "U2+U12
 # !res & !u6 & !u5 & !u4 & !u3 &  u2 & !u1;                      "U2

u3 =
   !res & !u6 & !u5 & !u4       &  u2 & !u1 &  ack                "U2+U23
 # !res & !u6 & !u5 & !u4 &  u3       & !u1 &  ack                "U3+U23
 # !res & !u6 & !u5 & !u4 &  u3 & !u2 & !u1;                      "U3
```

```
u4 =
   !res & !u6 & !u5       &  u3 & !u2 & !u1 & !ack                "U3+U34
 # !res & !u6 & !u5 &  u4       & !u2 & !u1 & !ack & enw.q        "U4+U34
 # !res & !u6 & !u5 &  u4 & !u3 & !u2 & !u1 &  enw.q;             "U4

u5 =
   !res & !u6       &  u4 & !u3 & !u2 & !u1 &  dro.q & enw.q      "U4+U45
 # !res & !u6 &  u5       & !u3 & !u2 & !u1 &  dro.q & enw.q      "U5+U45
 # !res & !u6 &  u5 & !u4 & !u3 & !u2 & !u1;                      "U5

u6 =
   !res &       u5 & !u4 & !u3 & !u2 & !u1 &  ack                 "U5+U56
 # !res &  u6      & !u4 & !u3 & !u2 & !u1 &  ack                 "U6+U56
 # !res &  u6 & !u5 & !u4 & !u3 & !u2 & !u1;                      "U6

ds      = !y6 & !y5 & !y4 & !y3 & !y2 &  y1      "Y1
        # !y6 & !y5 & !y4 &  y3 & !y2 & !y1      "Y3
        # !u6 & !u5 & !u4 & !u3 &  u2 & !u1      "U2
        # !u6 &  u5 & !u4 & !u3 & !u2 & !u1;     "U5

dri     =  y2           "Y2
        #  y3           "Y3
        #  y5           "Y5
        #  u4 & !u3     "U4+U45
        #  u5           "U5+U45
        #  u6;          "U6+U56


"latch (read) data from SMI in data register, DREG.

lrd     = enr.q & (y1 & !dro.q # y3 &  dro.q) # lwd;

"output enable (read) data from DREG register to submodule

oerd    =  enr.q # enwd;

"clock write data from ethernet controllers to data register

clkwd   = brdy.q & enw.q;

"output enable (write) data from DREG register to SMI

!oewd    = enw.q # dsi & enrd;

"###########################################################################
"                       definition of counter
"###########################################################################
declarations;

c0      node    istype'reg_t,buffer';"least significant bit in counter
c1      node    istype'reg_t,buffer';
c2      node    istype'reg_t,buffer';
c3      node    istype'reg_t,buffer';
c4      node    istype'reg_t,buffer';"most significant bit in counter

c1u     node    istype'reg_d,buffer';"one clock each microsecond

d0      node    istype'reg_t,buffer';"least significant bit in counter
d1      node    istype'reg_t,buffer';
d2      node    istype'reg_t,buffer';
d3      node    istype'reg_t,buffer';"most significant bit in counter

equations;

c0.c    = clk;
c1.c    = clk;
c2.c    = clk;
c3.c    = clk;
c4.c    = clk;

c1u.c   = clk;

d0.c    = clk;
d1.c    = clk;
d2.c    = clk;
d3.c    = clk;

c0.ar    = res;
c1.ar    = res;
c2.ar    = res;
c3.ar    = res;
c4.ar    = res;

c1u.ar   = res;

d0.ar    = res;
d1.ar    = res;
d2.ar    = res;
d3.ar    = res;


c0.t    =                                !t.q # c0.q & t.q;
c1.t    =                        c0.q & !t.q # c1.q & t.q;
c2.t    =                c1.q & c0.q & !t.q # c2.q & t.q;
c3.t    =        c2.q & c1.q & c0.q & !t.q # c3.q & t.q;
c4.t    = c3.q & c2.q & c1.q & c0.q & !t.q # c4.q & t.q;
c1u.d   = c4.q & c3.q & c2.q & c1.q & c0.q & !t.q;

d0.t    =                        c1u.q & !t.q # d0.q & t.q;
d1.t    =                d0.q & c1u.q & !t.q # d1.q & t.q;
d2.t    =        d1.q & d0.q & c1u.q & !t.q # d2.q & t.q;
d3.t    = d2.q & d1.q & d0.q & c1u.q & !t.q # d3.q & t.q;

t.t     = d3.q & d2.q & d1.q & d0.q &  !t.q;
```

```
declarations;

HOLD     = [hold1, hold0];

STAT     = [n2, n1, n0];
STAT0    = [n0b, n0a];
HOLDA    = [holda1, holda0];

STS      = [n4, n3];

STR      = [y6, y5, y4, y3, y2, y1];

T0       = ^b000000;
Y1       = ^b000001;
Y2       = ^b000010;
Y3       = ^b000100;
Y4       = ^b001000;
Y5       = ^b010000;
Y6       = ^b100000;

STW      = [u6, u5, u4, u3, u2, u1];

U1       = ^b000001;
U2       = ^b000010;
U3       = ^b000100;
U4       = ^b001000;
U5       = ^b010000;
U6       = ^b100000;

"Input signals from SMISCU

SMI      = [enrd,enwd,lwd];

@include'contst1.abl';"test vectors for arbitration.
@include'contst2.abl';"test vectors for SMI read cycles
@include'contst3.abl';"test vectors for SMI write cycles

end tpe200;
```

```
"controller for 4 port ethernet submodule for baio
"kan 31/5-1994

"changed to 2 channels.kan 28/7-1995
"test input pin removed
"mask register changed.
"interrupt pins removed for controller 2 and 3.
"interrupt register changed.
"channel atention and port removed for channel 2 and 3.
"state Y8 added between Y1 and Y2/Y5
"jump from Y1 to T0 if !pas & T

module tpe101;

declarations;

tpe101 device 'mach220a';

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


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

"input signals from SMI

!pas              pin 17  ;"physical address strobe from vsb
 read             pin 51  ;"read signal from vsb. Latched with address
!ds               pin 20  ;"data strobe from vsb

ad27              pin 46  ;"address bit 27
ad26              pin 26  ;"address bit 26
ad25              pin 25  ;"address bit 25

!id0              pin  9  ;"ID from SMI connector
!id1              pin  7  ;"ID from SMI connector

"There is an access to the submodule when

"id(1:0)          ad(27:25)
"   0 1              1 0 0        submodule 1. Base address 0x0800 0000
"   1 0              1 0 1        submodule 2. Base address 0x0a00 0000
"   1 1              1 1 0        submodule 3. Base address 0x0c00 0000

ad10              pin  3  ;"address bit 10
ad9               pin  2  ;"address bit  9
ad8               pin 16  ;"address bit  8
ad7               pin 49  ;"address bit  7. these bits ared decoded to give
                          ;"these bits are  decoded to give various
                          ;"chip select.

"input signals from  submodule.

T                 pin 43  ;"input signal from delay line.
                           "the delay line is driven by the output signa
                           "t from this PAL.

clk               pin 15  ;"clock input pin connected to 30 MHz

!resi             pin 54  ;"reset input

t                 pin 28  ;"output signal to delay line;


"############################################################################
"                input signals from ethernet controllers
"############################################################################
 int0             pin 50  ;"interrupt from controller 0
 int1             pin  5  ;"interrupt from controller 1

!pchk0            pin 44  ;"parity error from controller 0
!pchk1            pin  4  ;"parity error from controller 0

"############################################################################
"                        nodes
"############################################################################

y8                node    ;"state variable
y7                node    ;"state variable
y6                node    ;"state variable
y5                node    ;"state variable
y4                node    ;"state variable
y3                node    ;"state variable
y2                node    ;"state variable
y1                node    ;"state variable

cm                node    ;"match signal. access to submodule

rdyi              node    istype'reg_t,buffer';"internal ready signal
rdys              pin  21 istype'reg_t,buffer';"rdy synchronized
reqi              node    istype'com,buffer';
led               pin     istype'reg_t,invert';

n2                node    istype'reg_d,buffer';
n1                node    istype'reg_d,buffer';
n0                node    istype'reg_d,buffer';

"the state variables defines the following states:

"T0 = !y8&!y7&!y6&!y5&!y4&!y3&!y2&!y1;
"Y1 = !y8&!y7&!y6&!y5&!y4&!y3&!y2& y1;      address handshake
"Y18=  y8&!y7&!y6&!y5&!y4&!y3&!y2& y1;
"Y8 =  y8&!y7&!y6&!y5&!y4&!y3&!y2&!y1;
"Y82=  y8&!y7&!y6&!y5&!y4&!y3& y2&!y1;
```

```
"Y2 = !y8&!y7&!y6&!y5&!y4&!y3& y2&!y1;      write cycle wait for data
"Y23= !y8&!y7&!y6&!y5&!y4& y3& y2&!y1;
"Y3 = !y8&!y7&!y6&!y5&!y4& y3&!y2&!y1;      start write cycle; set request
"Y34= !y8&!y7&!y6&!y5& y4& y3&!y2&!y1;
"Y4 = !y8&!y7&!y6&!y5& y4&!y3&!y2&!y1;      data handshake to SMI
"Y85=  y8&!y7&!y6& y5&!y4&!y3&!y2&!y1;
"Y5 = !y8&!y7&!y6& y5&!y4&!y3&!y2&!y1;      start read cycle; set request
"Y56= !y8&!y7& y6& y5&!y4&!y3&!y2&!y1;
"Y6 = !y8&!y7& y6&!y5&!y4&!y3&!y2&!y1;      wait for read data
"Y17= !y8& y7& y6&!y5&!y4&!y3&!y2& y1;
"Y7 = !y8& y7&!y6&!y5&!y4& y3& y2&!y1;

resn    node    istype'reg_t,buffer';   "programmed reset.
                                        "control register bit 0

inten   node    istype'reg_t,buffer';   "interrupt enable
                                        "control register bit 7

cnt2    node    istype'reg_t,buffer';   "control register bit 2
cnt3    node    istype'reg_t,buffer';   "control register bit 3
cnt4    node    istype'reg_t,buffer';   "control register bit 4

csc     node    istype'com,buffer';     "select control register
css     node    istype'com,buffer';     "select status register
csm     node    istype'com,buffer';     "select mask register
csi     node    istype'com,buffer';     "select interrupt register
oed     node    istype'com,buffer';     "output enable to PAL

rp      node    istype'reg_d,buffer';   "read pulse sync part
wp      node    istype'reg_d,buffer';   "write pulse sync part

rpx     node    istype'com,buffer';     "used to generate read pulses
wpx     node    istype'com,buffer';     "used to generate write pulses

intno1  node    istype'com,buffer';     "pending interrupt
intno2  node    istype'com,buffer';     "pending fault interrupt

"########################################################################
"               defintion of mask register ond other
"########################################################################

mask7   node    istype'reg_t,buffer';   "mask register bit 7
mask6   node    istype'reg_t,buffer';   "mask register bit 6
mask5   node    istype'reg_t,buffer';   "mask register bit 5
mask4   node    istype'reg_t,buffer';   "mask register bit 4
mask3   node    istype'reg_t,buffer';   "mask register bit 3
mask2   node    istype'reg_t,buffer';   "mask register bit 2
mask1   node    istype'reg_t,buffer';   "mask register bit 1
mask0   node    istype'reg_t,buffer';   "mask register bit 0

stpe1   pin  55 istype'com,buffer';     "select tpe on channel 1
stpe0   pin  37 istype'com,buffer';     "select tpe on channel 0
enpow1  pin  38 istype'com,buffer';     "enable power to coax channel 1
enpow0  pin  11 istype'com,buffer';     "enable power to coax channel 0
aport1  pinistype'com,buffer';      "automatic port selection on channel 1
aport0  pinistype'com,buffer';      "automatic port selection on channel 0

pint1   node    istype'reg_t,buffer';   "parity error from cont. 1
pint0   node    istype'reg_t,buffer';   "parity error from cont. 0

"########################################################################
"                       output signals
"########################################################################

"output signals to SMI;

ac      pin 36 istype'com,invert';      "address complete
asack   pin  6 istype'com,buffer';      "address acknowledge
ack     pin 24 istype'com,buffer';      "data handshake.
int     pin 29 istype'com,buffer';      "interrupt request


"output signals to registers.

latchrd pin 56  istype'reg_d,buffer';   "latch read data
                                        "generated from sync part
                                        "signal to DREG

"data i/o pins for control and status to SMI

dat7    pin 30;         "data bit 7
dat6    pin 31;         "data bit 6
dat5    pin 32;         "data bit 5
dat4    pin 48;         "data bit 4
dat3    pin 33;         "data bit 3
dat2    pin 12;         "data bit 2
dat1    pin 13;         "data bit 1
dat0    pin 14;         "data bit 0

ale     pin 23  istype'com,buffer'; "address latch enable. High signal
                                    "opens address latch.

enrd    pin 40  istype'com,buffer'; "output enable read data to SMI
                                    "generated from async part. to CONT1
enwd    pin 22  istype'com,buffer'; "output enable write data from SMI
                                    "generated from sync part. to CONT1
lwd     pin 41  istype'com,buffer'; "latch write data. async part. to CONT1

"########################################################################
"                       pins for parity check/generation
"########################################################################

oddi    pin 45;         "input from parity generator 74x280

par     pin 10;         "parity bit for data. I/O pin
```

```
oddp    pin 47  istype'reg_t,buffer'; "use odd parity;

flten   node    istype'reg_t,buffer'; "enable parity fault interrupt

"error in databit( 7: 0)
err0    node    istype'reg_t,buffer';

        "once set the error flags is kept until they are
        "until cleared by a write cycle to the status register

"#########################################################################
"                    pins read and write pulses
"#########################################################################

rpid    pin 62  istype'reg_d,invert';"read pulse to ID-PROM
rpvers  pin 63  istype'reg_d,invert';"read pulse to VERS-PROM

ca0     pin 65  istype'reg_d,invert';"channel attention to ethernet 0
ca1     pin 64  istype'reg_d,invert';"channel attention to ethernet 1

port0   pin 67  istype'reg_d,invert';"port signal to ethernet 0
port1   pin 66  istype'reg_d,invert';"port signal to ethernet 1

"#########################################################################
"                    equations for state variables
"#########################################################################

equations;

y1  = !resi & (
          !y8&!y7&!y6&!y5&!y4&!y3&!y2     & pas & !T "(T0+Y1)
    #     !y8&!y7&!y6&!y5&!y4&!y3&!y2& y1       & !T "Y1
    #     !y8&!y7&!y6&!y5&!y4&!y3&!y2& y1 & pas);    "Y1

y2  = !resi & (
           y8&!y7&!y6&!y5&!y4&!y3    &!y1 & !read & !T "(Y8+Y82)
    #         !y7&!y6&!y5&!y4&!y3& y2&!y1 & !read & !T "(Y82+Y2)
    #     !y8&!y7&!y6&!y5&!y4&!y3& y2&!y1);            "Y2

y3  = !resi & (
          !y8&!y7&!y6&!y5&!y4    & y2&!y1 & ds        "(Y2+Y23)
    #     !y8&!y7&!y6&!y5&!y4& y3    &!y1 & ds        "(Y23+Y3)
    #     !y8&!y7&!y6&!y5&!y4& y3&!y2&!y1);           "Y3

y4  = !resi & (
          !y8&!y7&!y6&!y5    & y3&!y2&!y1 & !rdyi.q      "(Y3+Y34)
    #     !y8&!y7&!y6&!y5& y4    &!y2&!y1 & !rdyi.q & pas"(Y34+Y4)
    #     !y8&!y7&!y6&!y5& y4    &!y2&!y1 & !rdyi.q & ds "(Y34+Y4)
    #     !y8&!y7& y6&!y5    &!y3&!y2&!y1 &  rdyi.q & ds "(Y6+Y64)
    #     !y8&!y7&    !y5& y4&!y3&!y2&!y1 &  rdyi.q & ds "(Y64+Y4)
    #     !y8&!y7&!y6&!y5& y4&!y3&!y2&!y1 & pas           "Y4
    #     !y8&!y7&!y6&!y5& y4&!y3&!y2&!y1 & ds );         "Y4

y5  = !resi & (
           y8&!y7&!y6    &!y4&!y3&!y2&!y1 & read & !T"(Y8+Y85)
    #         !y7&!y6& y5&!y4&!y3&!y2&!y1 & read & !T"(Y85+Y5)
    #     !y8&!y7&!y6& y5&!y4&!y3&!y2&!y1);          "Y5

y6  = !resi & (
          !y8&!y7    & y5&!y4&!y3&!y2&!y1 & !rdyi.q      "(Y5+Y56)
    #     !y8&!y7& y6    &!y4&!y3&!y2&!y1 & !rdyi.q      "(Y56+Y6)
    #     !y8&!y7& y6&!y5&!y4&!y3&!y2&!y1);              "Y6

y7  = !resi & (
          !y8&    !y6&!y5&!y4&!y3&!y2& y1 & pas & T & !cm "(Y1+Y17)
    #     !y8& y7&!y6&!y5&!y4&!y3&!y2&     pas & T & !cm "(Y17+Y7)
    #     !y8& y7&!y6&!y5&!y4&!y3&!y2&!y1 & pas);         "Y7

y8  = !resi & (
            !y7&!y6&!y5&!y4&!y3&!y2& y1 & pas & T & cm & rdyi.q "(Y1+Y18)
    #      y8&!y7&!y6&!y5&!y4&!y3&!y2    & T & cm & rdyi.q       "(Y18+Y8)
    #      y8&!y7&!y6&!y5&!y4&!y3&!y2&!y1);                      "Y8

"#########################################################################
"                    equations for node variables
"#########################################################################


cm  = ad27 & !ad26 & !ad25 & !id1 &  id0        "submodule 1
    # ad27 & !ad26 &  ad25 &  id1 & !id0        "submodule 2
    # ad27 &  ad26 & !ad25 &  id1 &  id0;       "submodule 3

"#########################################################################
"                    equations for output to vsb
"#########################################################################

ack =   y4 & ds;                                "Y4

asack = !y1 & y8 # y2 # y3 # y5 # y6 # y4 & pas;

!ac =   y7 & pas # y2 # y8 # y3 # y5 # y6 # y4 & pas;

"#########################################################################
"                  equations for output to SCU module
"#########################################################################

"output to delay line

t =     y1;

ale = !asack & rdyi.q;      "latch open when high. Address latch is closed
                          "when asack is activated or rdyi.q deactivated.

lwd  =  y2 # y3;   "latch open when high
```

```
enrd   =  ds & read & (y6 # y4);        "oe read data to SMI

reqi   =  (y3 # y5);"internal request

"#########################################################################
"       equations  for internal registers
"#########################################################################

csc     = !ad10 & !ad9 &  ad8 &  ad7; "control register
css     = !ad10 & !ad9 &  ad8 & !ad7; "status register
csm     = !ad10 &  ad9 & !ad8 & !ad7; "mask register
csi     = !ad10 &  ad9 & !ad8 &  ad7; "pending interrupt register

oed     = !ad10 & !ad9 &  ad8 &  ad7  "control register
        # !ad10 & !ad9 &  ad8 & !ad7  "status register
        # !ad10 &  ad9 & !ad8 & !ad7  "mask register
        # !ad10 &  ad9 & !ad8 &  ad7; "pending interrupt register

dat7.oe = oed & rp.q;
dat6.oe = oed & rp.q;
dat5.oe = oed & rp.q;
dat4.oe = oed & rp.q;
dat3.oe = oed & rp.q;
dat2.oe = oed & rp.q;
dat1.oe = oed & rp.q;
dat0.oe = oed & rp.q;


intno1 =
        mask7.q & int1
      # mask6.q & int0;

intno2 =
        err0.q
      # mask5.q & pint1.q
      # mask4.q & pint0.q;

int   =  inten.q & intno1
      # flten.q & intno2;

dat7 =  css    & int
     #  csc    & inten.q
     #  csi    & stpe1.pin
     #  csm    & mask7.q;

dat6 =  css    & intno1
     #  csc    & flten.q
     #  csi    & stpe0.pin
     #  csm    & mask6.q;

dat5 =  css    & intno2
     #  csc    & oddp.q
     #  csi    & 0
     #  csm    & mask5.q;

dat4 =  css    & 0"test input pin removed
     #  csc    & cnt4.q
     #  csi    & 0
     #  csm    & mask4.q;

dat3 =  css    & 0
     #  csc    & cnt3.q
     #  csi    & int1
     #  csm    & mask3.q;

dat2 =  css    & 0
     #  csc    & cnt2.q
     #  csi    & int0
     #  csm    & mask2.q;

dat1 =  css    & 0
     #  csc    & led.q
     #  csi    & pint1.q
     #  csm    & mask1.q;

dat0 =  css    & err0.q
     #  csc    & resn.q
     #  csi    & pint0.q
     #  csm    & mask0.q;

"#########################################################################
"               equations  for parity check/generation
"#########################################################################

err0.c = clk;   "the error flag is clocked in all read an write cycles
err0.ar= resi;  "the error flag is cleared during reset

"parity bits are output in all read cycles to the module

par.oe = rp.q;          "all read cycles

par    = oddi & rp.q;   "The parity bit from the parity generator is
                        "output during all read cycles.

err0.t = !err0.q & (oddi $ par.pin) & wp.q      "check parity
       #  err0.q & css & wp.q;                  "clear error flag

"#########################################################################
"                       equations  for control register
"#########################################################################
resn.c  = clk;
led.c   = clk;
cnt2.c  = clk;
cnt3.c  = clk;
```

```
cnt4.c  = clk;
oddp.c  = clk;
flten.c = clk;
inten.c = clk;

resn.t  = csc & wp.q & ( dat0.pin & !resn.q  # !dat0.pin & resn.q );

led.t   = csc & wp.q & ( dat1.pin & !led.q   # !dat1.pin & led.q  );

cnt2.t  = csc & wp.q & ( dat2.pin & !cnt2.q  # !dat2.pin & cnt2.q );

cnt3.t  = csc & wp.q & ( dat3.pin & !cnt3.q  # !dat3.pin & cnt3.q );

cnt4.t  = csc & wp.q & ( dat4.pin & !cnt4.q  # !dat4.pin & cnt4.q );

oddp.t  = csc & wp.q & ( dat5.pin & !oddp.q  # !dat5.pin & oddp.q );

flten.t = csc & wp.q & ( dat6.pin & !flten.q # !dat6.pin & flten.q);

inten.t = csc & wp.q & ( dat7.pin & !inten.q # !dat7.pin & inten.q);

resn.ar  = resi;
led.ar   = resi;
cnt2.ar  = resi;
cnt3.ar  = resi;
cnt4.ar  = resi;
oddp.ar  = resi;
flten.ar = resi;
inten.ar = resi;

"###########################################################################
"                    equations  for mask register
"###########################################################################

mask7.c   = clk;
mask6.c   = clk;
mask5.c   = clk;
mask4.c   = clk;
mask3.c   = clk;
mask2.c   = clk;
mask1.c   = clk;
mask0.c   = clk;

mask7.ar  = resi;
mask6.ar  = resi;
mask5.ar  = resi;
mask4.ar  = resi;
mask3.ar  = resi;
mask2.ar  = resi;
mask1.ar  = resi;
mask0.ar  = resi;

mask7.t   = csm & wp.q & ( dat7.pin & !mask7.q  # !dat7.pin & mask7.q );
mask6.t   = csm & wp.q & ( dat6.pin & !mask6.q  # !dat6.pin & mask6.q );
mask5.t   = csm & wp.q & ( dat5.pin & !mask5.q  # !dat5.pin & mask5.q );
mask4.t   = csm & wp.q & ( dat4.pin & !mask4.q  # !dat4.pin & mask4.q );
mask3.t   = csm & wp.q & ( dat3.pin & !mask3.q  # !dat3.pin & mask3.q );
mask2.t   = csm & wp.q & ( dat2.pin & !mask2.q  # !dat2.pin & mask2.q );
mask1.t   = csm & wp.q & ( dat1.pin & !mask1.q  # !dat1.pin & mask1.q );
mask0.t   = csm & wp.q & ( dat0.pin & !mask0.q  # !dat0.pin & mask0.q );

"###########################################################################
"            equations  for interface selection
"###########################################################################

"interface for channel 0 is selected by mask(1:0)

"              a   e
"              p s n
"              o t p
"              r p o
"              t e w
"mask(1:0)     0 0 0

"     0 0      1 z 0     automatic port selection. Power is enabled
"     0 1      1 z 0     automatic port selection. Power is enabled
"     1 1      0 1 1     select tpe.               Power is disabled
"     1 0      0 0 0     select coax.              Power is enabled

aport0   = !mask1.q;
stpe0.oe =  mask1.q;
stpe0    =  mask0.q;
enpow0   =  mask1.q & mask0.q;

"during automatic port selection stpe is an input to the MACH

"interface for channel 1 is selected by mask(3:2)

aport1   = !mask3.q;
stpe1.oe =  mask3.q;
stpe1    =  mask2.q;
enpow1   =  mask3.q & mask2.q;

"###########################################################################
"            equations for interrupt register
"###########################################################################


pint1.c  = clk;
pint0.c  = clk;

pint1.ar  = resi;
pint0.ar  = resi;
```

```
pint1.t  = pchk1 & !pint1.q # csi & wp.q & pint1.q;
pint0.t  = pchk0 & !pint0.q # csi & wp.q & pint0.q;

"########################################################################
"                        synchronous part
"########################################################################

n2.ar = resi; "state variable
n1.ar = resi; "state variable
n0.ar = resi; "state variable

rdys.ap   = resi; "synchronized version of rdy
rdyi.ap   = resi; "rdyi is set by resi.

n2.c = clk; "state variable
n1.c = clk; "state variable
n0.c = clk; "state variable

rdyi.c   = clk;          "internal ready signal
rdys.c   = clk;          "synchronized version og rdy

rdyi.ar  = reqi;         "rdyi is cleared by reqi

wp.c   = clk;            "write pulse
rp.c   = clk;            "read pulse
latchrd.c  = clk;        "latch read data. Latch open when 1.
latchrd.ar = resi;       "cleared by reset

rdys.t = rdys.q & !rdyi.q; "clear rdys

declarations;

s0      = ^b000; "idle state;
s1      = ^b001;
s2      = ^b011;
s3      = ^b010;
s4      = ^b110;
s5      = ^b100;
s6      = ^b101;
s7      = ^b000;

ina  = [rdys.q, read];

state_diagram([n2,n1,n0]);

state s0:
"idle state

enwd = !read & !rdys.q;

rpx  =  read & !rdys.q;
wpx  = !read & !rdys.q;

case"          ys r
    (ina == [ 1, x]): s0;
    (ina == [ 0, x]): s1;
endcase;

state s1:

 enwd= !read;

rpx  =  read;
wpx  = !read;

goto s2;

state s2:

 enwd= !read;
latchrd.d = read;
rpx  =  read;
wpx  = !read;

goto s3;

state s3:

 enwd= !read;

goto s4;

state s4:

 enwd= !read;
rdyi.t = !rdyi.q; "set rdyi
rdys.t = !rdys.q; "set rdys

goto s0;

"########################################################################
"                 equations for read and write pulses
"########################################################################
equations;

rpid.c    = clk;"read pulse to ID-PROM
rpvers.c  = clk;"read pulse to VERS-PROM

ca0.c     = clk;"channel attention to ethernet 0
ca1.c     = clk;"channel attention to ethernet 1

port0.c   = clk;"port signal to ethernet 0
port1.c   = clk;"port signal to ethernet 1
```

```
rpid.ar     = resi;"read pulse to ID-PROM
rpvers.ar   = resi;"read pulse to VERS-PROM

ca0.ar      = resi;"channel attention to ethernet 0
ca1.ar      = resi;"channel attention to ethernet 1

port0.ar    = resi;"port signal to ethernet 0
port1.ar    = resi;"port signal to ethernet 1

rp.ar       = resi;
wp.ar       = resi;

rpid.d          = rpx & !ad10 & !ad9 & !ad8 & !ad7;
rpvers.d        = rpx & !ad10 & !ad9 & !ad8 &  ad7;

ca0.d           = wpx & !ad10 &  ad9 &  ad8 & !ad7;
port0.d         = wpx & !ad10 &  ad9 &  ad8 &  ad7;
ca1.d           = wpx &  ad10 & !ad9 & !ad8 & !ad7;
port1.d         = wpx &  ad10 & !ad9 & !ad8 &  ad7;

rp.d            = rpx;
wp.d            = wpx;

declarations;

ID = [id1, id0];

MA = [ad27, ad26, ad25];

AD =  [ad10, ad9, ad8, ad7];

DAT = [dat7, dat6, dat5, dat4, dat3, dat2, dat1, dat0];

TIL   = [y8, y7, y6, y5, y4, y3, y2, y1];

T0 = ^b00000000;
Y1 = ^b00000001;
Y2 = ^b00000010;
Y3 = ^b00000100;
Y4 = ^b00001000;
Y5 = ^b00010000;
Y6 = ^b00100000;
Y7 = ^b01000000;
Y8 = ^b10000000;

I30 = [int1, int0, pchk1, pchk0];

PULS = [!rpid, !rpvers];
CAP = [!ca1,!ca0,!port1,!port0];

"id :   id1, id0 (ID)
"ma :   ad27, ad26, ad25 (MA)
"adr:   register address
"dat:   write data
"p:     parity bit for write data
"o:     input from parity checker
"i:     interrupt pin
"e:     test input pin
"cap:   ca(3:0), port(3:0)
"i30:   int(3:0), pchk(3:0)
"rpl:   rpid, rpvers (PULS)
"q:     oddp
"#########################################################################
"                  test vectors for no access
"#########################################################################

@const id       = ^b00;
@const ma       = ^b000;
@const adr      = ^h0;
@const dat      = ^h00;
@const p        = ^b0;
@const o        = ^b0;
@const i        = ^b0;
@const e        = ^b0;
@const cap      = ^h00;
@const i30      = ^h00;
@const rpl      = ^b00;
@const q        = ^b0;
@include 'smitstna.abl'; "test no access  4

@const id       = ^b01;
@const ma       = ^b101;
@include 'smitstna.abl'; "test no access  8

@const ma       = ^b110;
@include 'smitstna.abl'; "test no access  12

@const ma       = ^b000;
@include 'smitstna.abl'; "test no access  16


@const id       = ^b10;
@const ma       = ^b100;
@include 'smitstna.abl'; "test no access  20

@const ma       = ^b110;
@include 'smitstna.abl'; "test no access  24

@const ma       = ^b001;
@include 'smitstna.abl'; "test no access  28


@const id       = ^b11;
@const ma       = ^b111;
@include 'smitstna.abl'; "test no access  32
```

```
@const ma       = ^b100;
@include 'smitstna.abl'; "test no access  36

@const ma       = ^b010;
@include 'smitstna.abl'; "test no access  40

"########################################################################
"                test vectors for control register
"########################################################################


@const id       = ^b01;
@const ma       = ^b100;
@const adr      = ^h3;          "control register address

@const dat      = ^h00;
@include 'smitstw.abl'; "write ^h00       50
@include 'smitstr.abl'; "read  ^h00       59

@const dat      = ^h01;
@include 'smitstw.abl'; "write ^h01       69
@include 'smitstr.abl'; "read  ^h01       78

@const dat      = ^h02;
@include 'smitstw.abl'; "write ^h02       88
@include 'smitstr.abl'; "read  ^h02       97

@const dat      = ^h04;
@include 'smitstw.abl'; "write ^h04      107
@include 'smitstr.abl'; "read  ^h04      116

@const dat      = ^h08;
@include 'smitstw.abl'; "write ^h08      126
@include 'smitstr.abl'; "read  ^h08      135

@const dat      = ^h10;
@include 'smitstw.abl'; "write ^h10      145
@include 'smitstr.abl'; "read  ^h10      154

@const q        = ^b1;
@const dat      = ^h20;
@include 'smitstw.abl'; "write ^h20      164
@include 'smitstr.abl'; "read  ^h20      173

@const q        = ^b0;
@const dat      = ^h40;
@include 'smitstw.abl'; "write ^h40     183
@include 'smitstr.abl'; "read  ^h40     192

@const dat      = ^h80;
@include 'smitstw.abl'; "write ^h80     202
@include 'smitstr.abl'; "read  ^h80     211

"########################################################################
"                test vectors for mask register
"########################################################################


@const id       = ^b10;
@const ma       = ^b101;
@const adr      = ^h4;          "mask register address

@const dat      = ^h00;
@include 'smitstw.abl'; "write ^h00    221
@include 'smitstr.abl'; "read  ^h00    230

@const dat      = ^h01;
@include 'smitstw.abl'; "write ^h01    240
@include 'smitstr.abl'; "read  ^h01    249


@const dat      = ^h02;
@include 'smitstw.abl'; "write ^h02    259
@include 'smitstr.abl'; "read  ^h02    268


@const dat      = ^h04;
@include 'smitstw.abl'; "write ^h04    278
@include 'smitstr.abl'; "read  ^h04    287


@const dat      = ^h08;
@include 'smitstw.abl'; "write ^h08    297
@include 'smitstr.abl'; "read  ^h08    306


@const dat      = ^h10;
@include 'smitstw.abl'; "write ^h10    316
@include 'smitstr.abl'; "read  ^h10    325


@const dat      = ^h20;
@include 'smitstw.abl'; "write ^h20    335
@include 'smitstr.abl'; "read  ^h20    344


@const dat      = ^h40;
@include 'smitstw.abl'; "write ^h40    354
@include 'smitstr.abl'; "read  ^h40    363


@const dat      = ^h80;
@include 'smitstw.abl'; "write ^h80    373
@include 'smitstr.abl'; "read  ^h80    382
```

```
"#########################################################################
"          test vectors for channel attention and port
"#########################################################################
"cap:   ca(3:0), port(3:0)

@const id        = ^b11;
@const ma        = ^b110;
@const dat       = ^h00;                    "data is not used

@const adr       = ^h6;                     "ca0
@const cap       = ^h4;                     "
@include 'smitstw.abl';                     "392

@const adr       = ^h7;                     "port0
@const cap       = ^h1;                     "
@include 'smitstw.abl';                     "402

@const adr       = ^h8;                     "ca1
@const cap       = ^h8;                     "
@include 'smitstw.abl';                     "412

@const adr       = ^h9;                     "port1
@const cap       = ^h02;                    "
@include 'smitstw.abl';                     "422

@const adr       = ^ha;                     "ca2
@const cap       = ^h0;                     "
@include 'smitstw.abl';                     "432

@const adr       = ^hb;                     "port2
@const cap       = ^h0;                     "
@include 'smitstw.abl';                     "442

@const adr       = ^hc;                     "ca3
@const cap       = ^h0;                     "
@include 'smitstw.abl';                     "452

@const adr       = ^hd;                     "port3
@const cap       = ^h0;                     "
@include 'smitstw.abl';                     "462

@const adr       = ^h0;                     "rpid
@const cap       = ^h00;                    "
@const rpl       = ^h2;                     "
@include 'smitstx.abl';                     "471

@const adr       = ^h1;                     "rpvers
@const rpl       = ^h1;                     "
@include 'smitstx.abl';                     "480

@const rpl       = ^h0;                     "

"#########################################################################
"                test vectors for interrupt
"#########################################################################

@const id        = ^b11;
@const ma        = ^b110;

@const i         = ^b0;
@const adr       = ^h3;                     "clear control register
@const dat       = ^h00;
@include 'smitstw.abl';                     "write ^h00     490

@const adr       = ^h4;                     "clear mask register
@const dat       = ^h00;
@const i30       = ^h01;                    "set interrupt
@include 'smitstw.abl';                     "write ^h00     500

@const adr       = ^h5;                     "read pending interrupt
@const dat       = ^h01;
@const i30       = ^h00;                    "
@include 'smitstr.abl';                     "               509

@const adr       = ^h4;                     "set mask register
@const dat       = ^h10;
@include 'smitstw.abl';                     "               519

@const adr       = ^h2;                     "read status register
@const dat       = ^h20;
@include 'smitstr.abl';                     "               528

@const i         = ^b1;
@const adr       = ^h3;                     "
@const dat       = ^h40;                    "enable parity fault interupt
@include 'smitstw.abl';                     "write control  538

@const i         = ^b0;
@const adr       = ^h5;                     "clear pending interrupt
@const dat       = ^hff;
@include 'smitstw.abl';                     "               548

@const adr       = ^h5;                     "read pending interrupt
@const dat       = ^h00;
@include 'smitstr.abl';                     "               557

@const adr       = ^h2;                     "read status register
@const dat       = ^h00;
@include 'smitstr.abl';                     "               566

"#########################################################################

@const i         = ^b0;
```

```
@const adr      = ^h3;                  "clear control register
@const dat      = ^h00;
@include 'smitstw.abl';                 "write ^h00     576

@const adr      = ^h4;                  "clear mask register
@const dat      = ^h00;
@const i30      = ^h02;                 "set interrupt
@include 'smitstw.abl';                 "write ^h00     586

@const adr      = ^h5;                  "read pending interrupt
@const dat      = ^h02;
@const i30      = ^h00;                 "
@include 'smitstr.abl';                 "               595

@const adr      = ^h4;                  "set mask register
@const dat      = ^h20;
@include 'smitstw.abl';                 "               605

@const adr      = ^h2;                  "read status register
@const dat      = ^h20;
@include 'smitstr.abl';                 "               614

@const i        = ^b1;
@const adr      = ^h3;                  "
@const dat      = ^h40;                 "enable parity fault interupt
@include 'smitstw.abl';                 "write control  624

@const i        = ^b0;
@const adr      = ^h5;                  "clear pending interrupt
@const dat      = ^hff;
@include 'smitstw.abl';                 "               634

@const adr      = ^h5;                  "read pending interrupt
@const dat      = ^h00;
@include 'smitstr.abl';                 "               643

@const adr      = ^h2;                  "read status register
@const dat      = ^h00;
@include 'smitstr.abl';                 "               652

"#########################################################################

@const i        = ^b0;
@const adr      = ^h3;                  "clear control register
@const dat      = ^h00;
@include 'smitstw.abl';                 "write ^h00     834

@const adr      = ^h4;                  "clear mask register
@const dat      = ^h00;
@const i30      = ^h4;                 "set interrupt
@include 'smitstw.abl';                 "write ^h00     844

@const adr      = ^h5;                  "read pending interrupt
@const dat      = ^h4;
@include 'smitstr.abl';                 "               853

@const adr      = ^h4;                  "set mask register
@const dat      = ^h40;
@include 'smitstw.abl';                 "               863

@const adr      = ^h2;                  "read status register
@const dat      = ^h40;
@include 'smitstr.abl';                 "               872

@const i        = ^b1;
@const adr      = ^h3;                  "
@const dat      = ^h80;                 "enable interupt
@include 'smitstw.abl';                 "write control  882

@const i        = ^b0;
@const i30      = ^h00;                 "remove interrupt

@const adr      = ^h5;                  "read pending interrupt
@const dat      = ^h00;
@include 'smitstr.abl';                 "               891

@const adr      = ^h2;                  "read status register
@const dat      = ^h00;
@include 'smitstr.abl';                 "               900

"#########################################################################

@const i        = ^b0;
@const adr      = ^h3;                  "clear control register
@const dat      = ^h00;
@include 'smitstw.abl';                 "write ^h00     910

@const adr      = ^h4;                  "clear mask register
@const dat      = ^h00;
@const i30      = ^h8;                 "set interrupt
@include 'smitstw.abl';                 "write ^h00     920

@const adr      = ^h5;                  "read pending interrupt
@const dat      = ^h8;
@include 'smitstr.abl';                 "               929

@const adr      = ^h4;                  "set mask register
@const dat      = ^h80;
@include 'smitstw.abl';                 "               939

@const adr      = ^h2;                  "read status register
@const dat      = ^h40;
@include 'smitstr.abl';                 "               948

@const i        = ^b1;
```

```
@const adr      = ^h3;                  "
@const dat      = ^h80;                 "enable interupt
@include 'smitstw.abl';                 "write control  958

@const i        = ^b0;
@const i30      = ^h00;                 "remove interrupt

@const adr      = ^h5;                  "read pending interrupt
@const dat      = ^h00;
@include 'smitstr.abl';                 "               967

@const adr      = ^h2;                  "read status register
@const dat      = ^h00;
@include 'smitstr.abl';                 "               976

"#########################################################################
"                    test parity fault interrupt
"#########################################################################

@const i        = ^b0;                  "disable interrupt
@const adr      = ^h3;                  "clear control register
@const dat      = ^h00;
@include 'smitstw.abl';                 "write ^h00     1138

@const adr      = ^hff;                 "set parity fault interrupt
@const dat      = ^h00;
@const p        = ^b0;                  "parity bit
@const o        = ^b1;                  "input from parity checker
@include 'smitstw.abl';                 "write ^h00     1148

@const o        = ^h0;                  "input from parity checker
@const adr      = ^h2;                  "read status register
@const dat      = ^h21;
@include 'smitstr.abl';                 "               1157

@const i        = ^b1;
@const adr      = ^h3;                  "
@const dat      = ^h40;                 "enable interupt
@include 'smitstw.abl';                 "write control  1167

@const adr      = ^h2;                  "read status register
@const dat      = ^ha1;
@include 'smitstr.abl';                 "               1176

@const i        = ^b0;
@const adr      = ^h2;                  "clear interrupt
@const dat      = ^h00;                 "
@include 'smitstw.abl';                 "write status   1186

@const adr      = ^h2;                  "read status register
@const dat      = ^h00;
@include 'smitstr.abl';                 "               1195

"#########################################################################

@const adr      = ^h3;                  "clear control register
@const dat      = ^h00;
@include 'smitstw.abl';                 "write ^h00     1205

@const adr      = ^hff;                 "set parity fault interrupt
@const dat      = ^h00;
@const p        = ^b1;                  "parity bit
@const o        = ^b0;                  "input from parity checker
@include 'smitstw.abl';                 "write ^h00     1215

@const p        = ^h0;                  "input from parity checker
@const adr      = ^h2;                  "read status register
@const dat      = ^h21;
@include 'smitstr.abl';                 "               1224

@const i        = ^b1;
@const adr      = ^h3;                  "
@const dat      = ^h40;                 "enable interupt
@include 'smitstw.abl';                 "write control  1234

@const adr      = ^h2;                  "read status register
@const dat      = ^ha1;
@include 'smitstr.abl';                 "               1243

@const i        = ^b0;
@const adr      = ^h2;                  "clear interrupt
@const dat      = ^h00;                 "
@include 'smitstw.abl';                 "write status   1253

@const adr      = ^h2;                  "read status register
@const dat      = ^h00;
@include 'smitstr.abl';                 "               1262

"#########################################################################

@const adr      = ^h3;                  "clear control register
@const dat      = ^h00;
@include 'smitstw.abl';                 "write ^h00     1272

@const adr      = ^hff;                 "set no parity fault interrupt
@const dat      = ^h00;
@const p        = ^b1;                  "parity bit
@const o        = ^h1;                  "input from parity checker
@include 'smitstw.abl';                 "write ^h00     1282

@const adr      = ^h2;                  "read status register
@const dat      = ^h00;
@include 'smitstr.abl';                 "               1291

@const i        = ^b0;
```

```
@const adr      = ^h3;                  "
@const dat      = ^h40;                 "enable interupt
@include 'smitstw.abl';                 "write control  1301

@const adr      = ^h2;                  "read status register
@const dat      = ^h00;
@include 'smitstr.abl';                 "               1310

@const adr      = ^h2;                  "clear interrupt
@const dat      = ^h00;                 "
@include 'smitstw.abl';                 "write status   1320

@const adr      = ^h2;                  "read status register
@const dat      = ^h00;
@include 'smitstr.abl';                 "               1329

end tpe101;
```

# Beskrivelse.

| | |
|---|---|
| name: | TPE301 |
| dato: | 94.07.14 |
| forfatter: | kan |
| version: | 1 |
| id: | xx-xxxx |
| status: | Preliminary |

## 1. Beskrivelse af styring.

Det følgende er en beskrivelse af styringen af TPE301 med henvisning til tilstandsdiagrammer. På side 1 er vist en oversigtstegning over styringen.

## 2. Abitrering.

Arbitreringen udføres af sekvensmaskiner, som kan deles i to grupper:

A: Fire synkrone sekvensmaskiner, en for hver controller, som tilsammen implementerer en round robin arbitrering.

B: En synkron/asynkron sekvensmaskine, der er placeret tæt ved SMI. Denne modtager REQ fra A, udfører arbitreringen på SMI og sender GAH (go ahead) til A.

## 2.1. Sekvensmaskinerne A.

Tilstandsdiagrammet findes på side 3.

st0: Hviletilstand, som forlades, når controller sender HOLD, og controller har prioritet. I denne tilstand sendes REQ til maskine B, når controller sender HOLD. Signalet GAH fra maskine B indgår i prioritet. ROUND sættes, når controller sender HOLD.

st1: I denne tilstand udføres accessen fra controller. Ved indhop til tilstanden sættes en timer. Der sendes HOLDA til controller, så længe HOLD er aktiveret, indtil timer er udløbet. Maskinen forbliver i tilstanden, indtil controller har fjernet HOLD. ROUND holdes, indtil HOLD er fjernet.

st2: Ved indhop resettes BUSY til SMI, så en ny arbitrering kan foretages. Maskinen forbliver i tilstand st2, indtil alle sekvensmaskiner A har fjernet ROUND. Herefter hopper alle maskiner til st0.

## 2.2. Sekvensmaskine B.

Tilstandsdiagrammet findes på side 2.

Sekvensmaskinen modtager REQ fra A-maskinerne og sender GAH til disse, når arbitreringen er udført. Hvis der ikke er REQ fra A-maskinerne, sendes BGOUT, når BGIN modtages.

BGIN er asynkron, mens REQ er synkron.

s0: Vent på REQ eller BGIN.

s1: I denne tilstand er der sendt request til SMI. Der ventes på BGIN aktiv og PAS inaktiv. Når dette indtræffer sættes GAH, og der hoppes til s2 på GAH.

s2: Ved indhop i denne tilstand sættes BUSY i SMI som tegn på, at en ny busmaster er valgt. Maskinen forbliver i s2, indtil BGIN er inaktiv. For at sikre, at maskinen ser BGIN inaktiv, anvendes en asynkron sekvensmaskine, som genererer signalet Y.

s3: I denne tilstand er BGIN modtaget, uden kortet har sendt request til SMI. BGIN sendes videre til næste kort som signalet BGOUT. Maskinen forbliver i denne tilstand, indtil BGIN er inaktiv. For at detektere BGIN inaktiv anvendes igen en asynkron sekvensmaskine, som genererer signalet Z. Z anvendes også ved generering af BGOUT.

## 3. Buscycle på SMI.

Når arbitreringen er udført, har en og kun en ethernetcontroller HOLDA. Denne controller vil herefter udføre en eller flere burst cycles. Controlleren starter en cycle med signalet ADS. Den modtager data, når man sender BRDY, burst ready, til den. Cyclen afsluttes, når controller sender BLAST, burst last, samtidigt med at den modtager sidste data på BRDY.

Alle signaler er synkrone med klokken.

Cycles fra controller skal oversættes til de handshakede cycles på SMI. En cycle på SMI består af adressefase og datafase. I datafasen overføres et eller flere ord. Styringen af datafasen er af afgørende betydning for overførselshastigheden.

Styringen er opdelt i en synkron sekvensmaskine, som håndterer både adresse- og datafase, og som kommunikerer med ethernet controller, og to asynkrone sekvensmaskiner: en til læsning og en anden til skrivning. Disse enables af henholdsvis ENR og ENW.

Den synkrone sekvensmaskine er vist på side 4 og de to asynkrone på side 5.

Kommunikationen mellem den synkrone og de asynkrone maskiner sker ved anvendelse af signalerne DRO til den asynkrone maskine og DRI fra den asynkrone maskine. DRO og DRI er tofasede: Det er ikke deres niveauer, der overfører information, men derimod skift. Ved læsning er princippet følgende:

Når den asynkrone maskine har læst data på SMI, ændrer den værdien af DRI. Når den synkrone har overført data til ethernetcontroller, ændrer den værdien af DRO, o.s.v.

For at opnå størst mulig overførselshastighed anvendes yderligere read ahead og bufrede skrivninger.

Read ahead: Så snart læs data er latched i DREG, startes en ny læsning på SMI, mens data overføres til ethernet controller. Dette medfører, at der læses for langt, hvorved blokgrænser kan overskrides. Den asynkrone sekvensmaskine til læsning skal derfor reagere på signalet AC, address complete, fra SMI. Reaktionen er: stop handshake og vent på ENR inaktiv.

Bufrede skrivninger: Så snart skrivdata er klokket i DREG, sendes BRDY til ethernetcontroller, som udlæser næste ord, mens der handshakes på SMI. Dette medfører at SMI stadigt er optaget, efter cyclen fra ethernet controller er afsluttet. Ethernet controller kan endog starte en ny cycle, ADS, mens SMI er optaget af skrivningen.

### 3.1. Synkron sekvensmaskine.

st0: vent på cycle fra ethernet. Når CYCLE er sat, åbnes adresselatch og enables til SMI kombinatorisk. Ved udhop fra st0 på CYCLE lukkes adresselatch, men output enable bevares. Ved udhop sættes PAS, og DRO sættes til 0 ved læsning og 1 ved skrivning. De asynkrone sekvensmaskiner er holdt i ro, DRI = 0, da både ENR og ENW er nul.

st1: Adressefase på SMI. Vent på synkroniseret version af ASACK, ASACKS. Adresselatch driver SMI, indtil ASACKS, kombinatorisk. Ved udhop sættes ENR eller ENW i henholdsvis læse eller skrive cycles. Disse signaler enabler den tilsvarende asynkrone sekvensmaskine.

st2: Datafase på SMI. I denne tilstand forblives, til sidste data er overført til/fra ethernet controller, BRDY*BLAST. I denne tilstand handshakes med den asynkrone sekvensmaskine med signalerne DRO, DRI, og med ethernetcontrolleren med signalet BRDY. BRDY sættes i en klokperiode, når DRO<>DRI. DRO ændres på den klok, hvor ethernet controller sampler BRDY. Ved læsning bevares værdien af DRO ved udhop fra st2. Ved skrivning ændres værdien af DRO ved udhop fra st2.

st3: Ved læsning er DRI<>DRO, hvorfor der umiddelbart hoppes til st0. Ved skrivning er ENW fortsat aktiveret, og der ventes til sidste data er handshaket med SMI, DRI<>DRO, hvorefter der hoppes til st0. Ved udhop deaktiveres PAS og ENW.

Bemærkninger: I datafasen, st2, sættes BRDY i en klokperiode, når DRI<>DRO. DRI er et asynkront signal. Alle øvrige signaler afhænger af BRDY og ikke af DRI, hvorved synkroniseringen er foretaget. I st3 anvendes DRIS, som er en synkroniseret udgave af DRI.

### 3.2. Asynkron sekvensmaskine til læsning.

T0: Hviletilstand, vent på ENR.

Y1: DS aktiveret. Vent på SMI (ACK) og vent på synkron sekvensmaskine (DRO). Ved start af læsning er DRO=0.

Y2: Læsdata er modtaget fra SMI (ACK). Signaler data klar til synkron maskine (DRI). DS deaktiveret. Vent på ACK deaktiveret.

Y5: Sidste læsdata er modtaget fra SMI (ACK*AC). Signaler data klar til synkron maskine (DRI) og vent på ACK og ENR deaktiveret.

Y3, Y4, Y6: svarer til henholdsvis Y1, Y2, Y5, men polariteten på DRO og DRI er omvendt.

### 3.3. Asynkron sekvensmaskine til skrivning.

T0: Vent på ENW.

U1: Vent på skrivdata fra synkronmaskine.

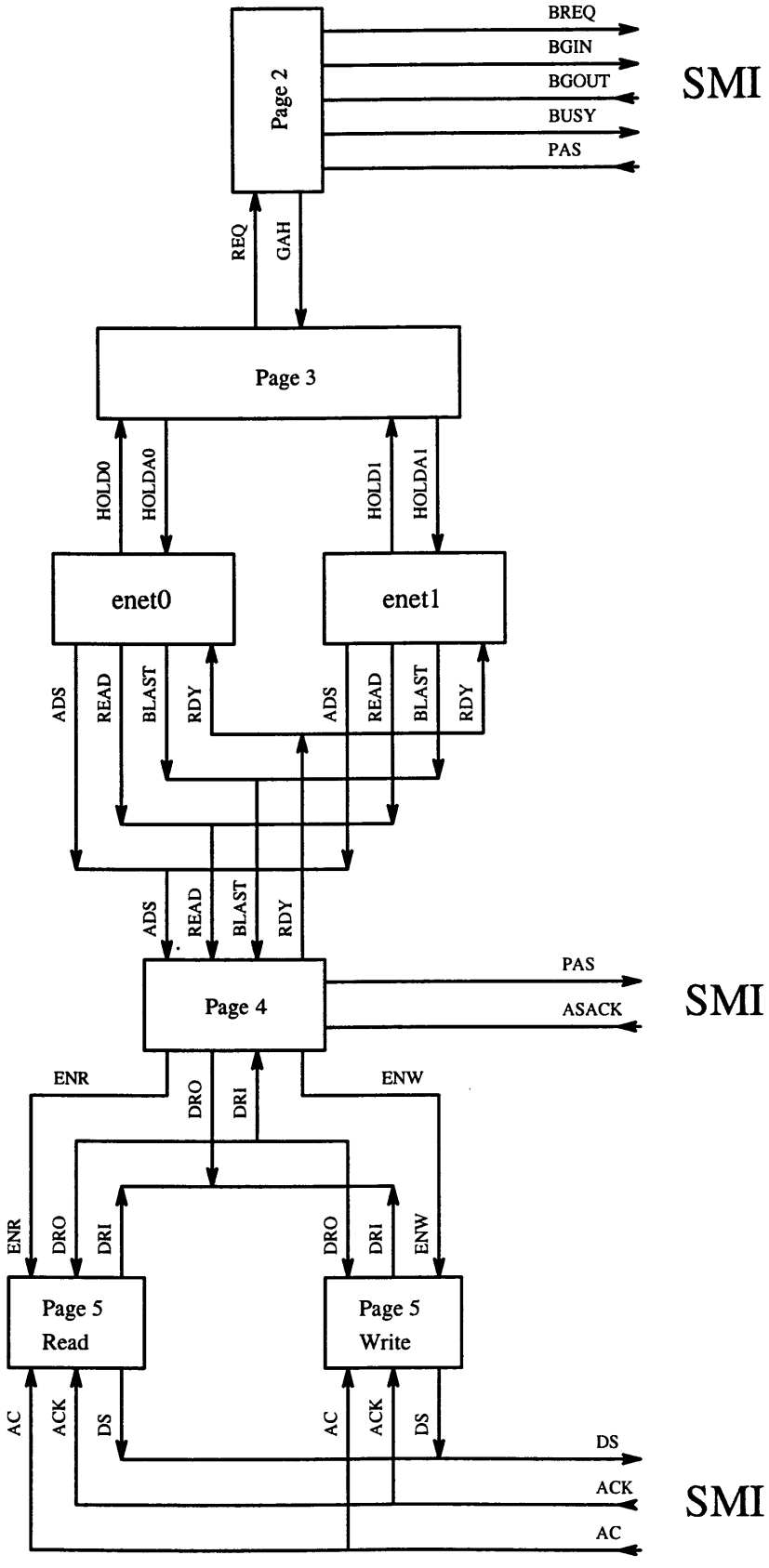
U2: Data på SMI; DS aktiv; Vent på ACK.

U3: Afslut handshake på SMI.

U4, U5, U6: svarer til henholdsvis U1, U2, U3 med DRo og DRI inverteret.

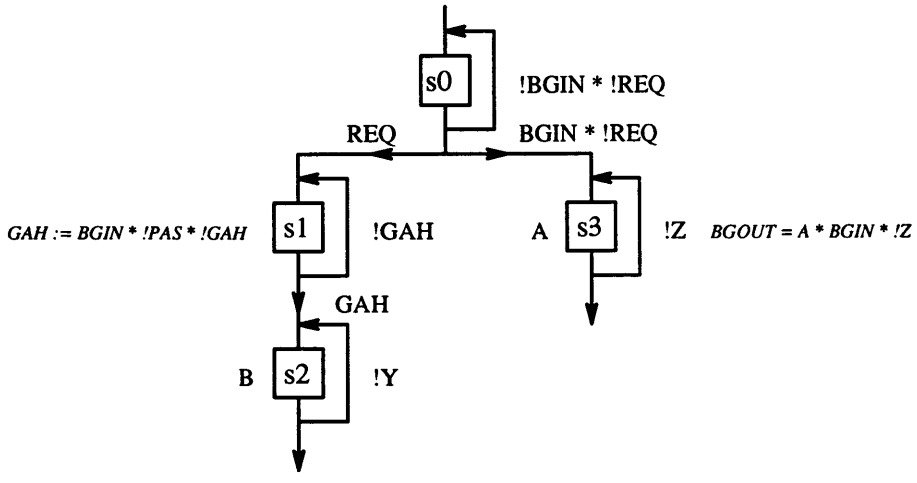
Dansk Data Elektronik A/S

| Control Overview | KAN | Page 1 |
|---|---|---|
| | 951106 | TPE301 |

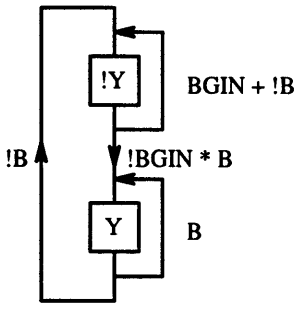


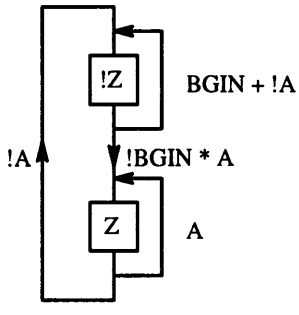
BGIN: asynkron
REQ: synkron

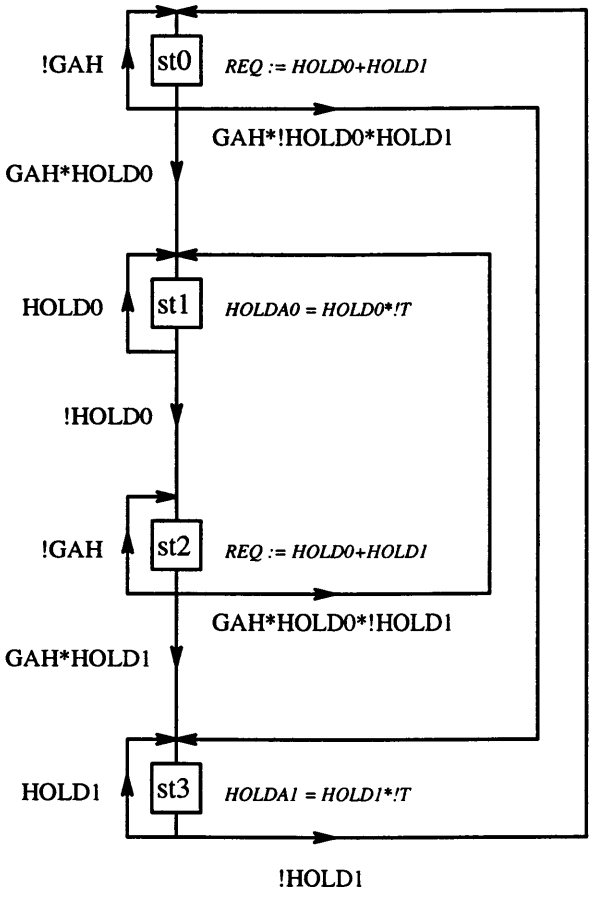
Synchronous State Machine

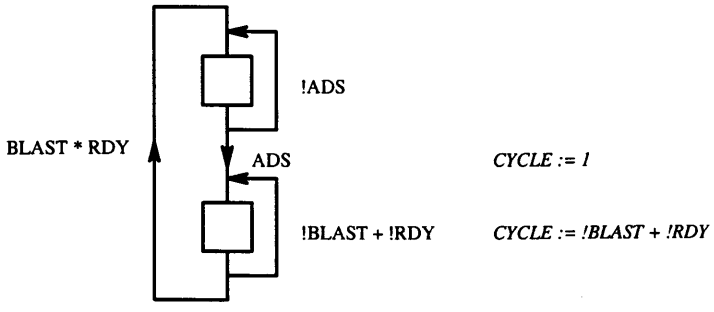


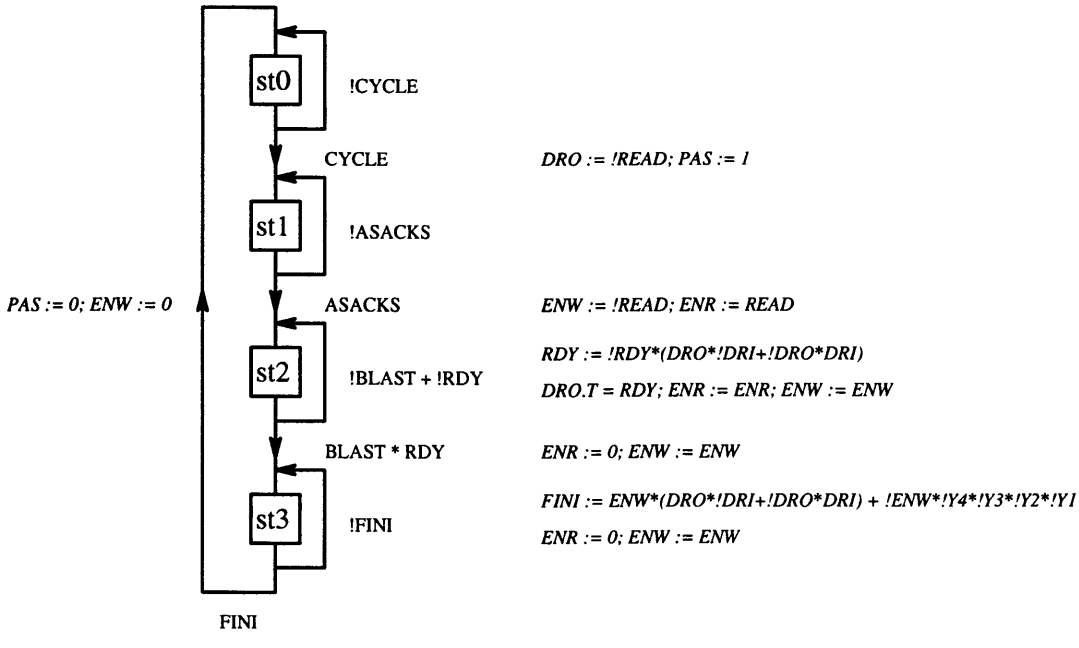

Asynchronous State Machines



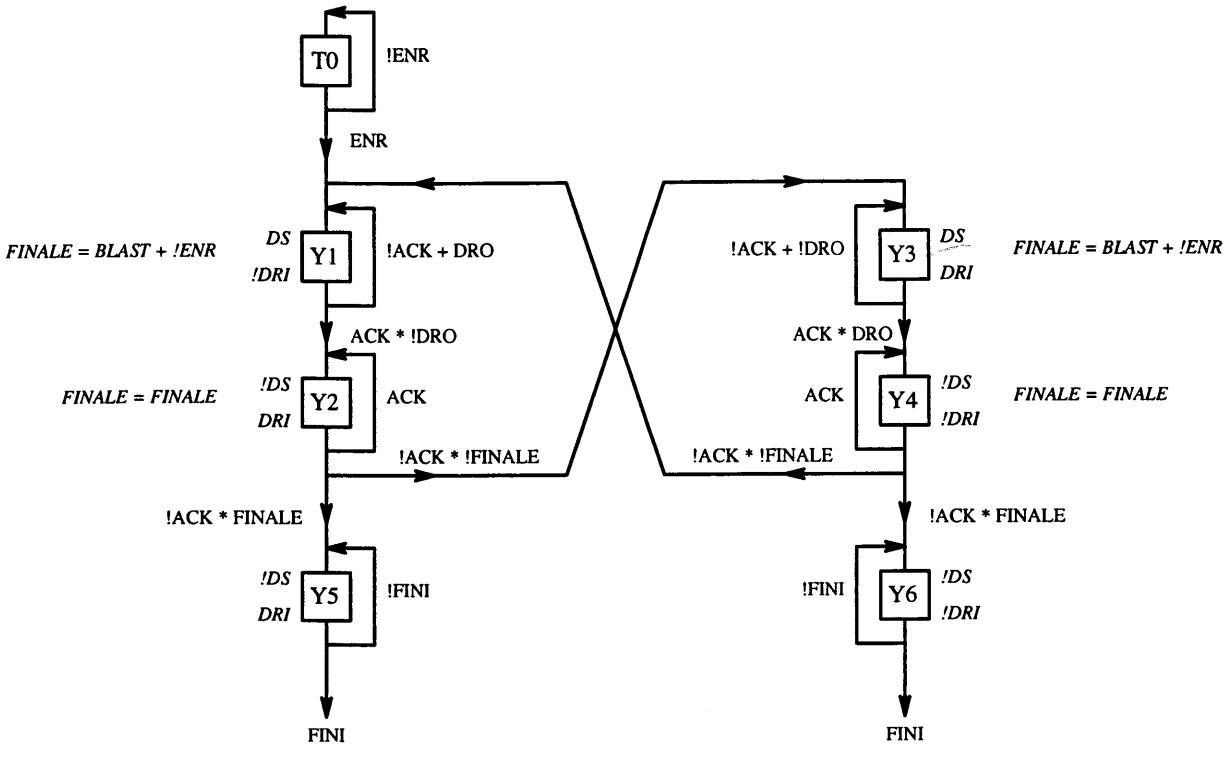




!GAH
st0
REQ := HOLD0+HOLD1
GAH*!HOLD0*HOLD1
GAH*HOLD0
HOLD0
st1
HOLDA0 = HOLD0*!T
!HOLD0
!GAH
st2
REQ := HOLD0+HOLD1
GAH*HOLD0*!HOLD1
GAH*HOLD1
HOLD1
st3
HOLDA1 = HOLD1*!T
!HOLD1

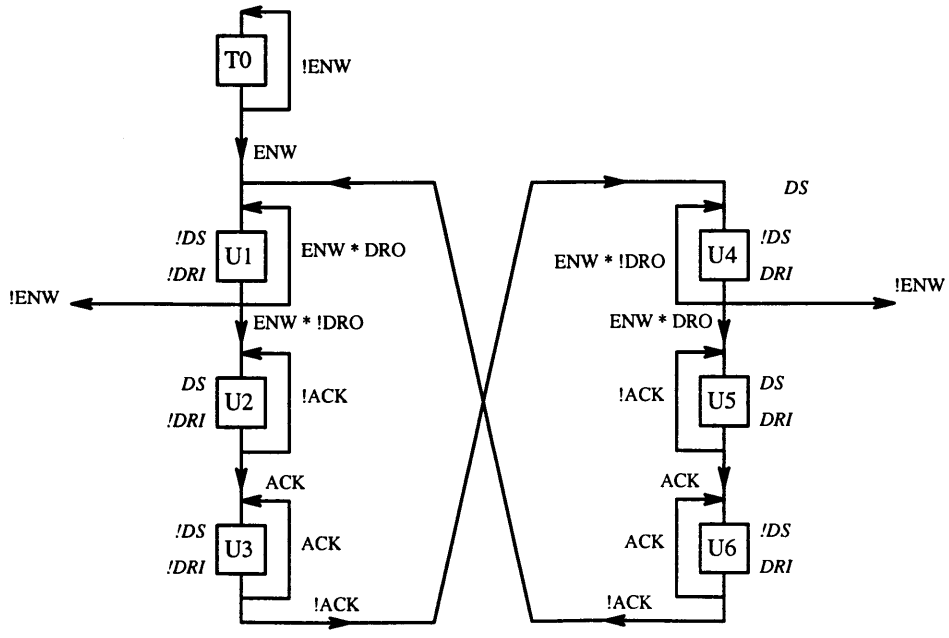

!ADS
BLAST * RDY
ADS
CYCLE := 1
!BLAST + !RDY
CYCLE := !BLAST + !RDY


st0
!CYCLE
CYCLE
DRO := !READ; PAS := 1
st1
!ASACKS
PAS := 0; ENW := 0
ASACKS
ENW := !READ; ENR := READ
RDY := !RDY*(DRO*!DRI+!DRO*DRI)
st2
!BLAST + !RDY
DRO.T = RDY; ENR := ENR; ENW := ENW
BLAST * RDY
ENR := 0; ENW := ENW
FINI := ENW*(DRO*!DRI+!DRO*DRI) + !ENW*!Y4*!Y3*!Y2*!Y1
st3
!FINI
ENR := 0; ENW := ENW
FINI

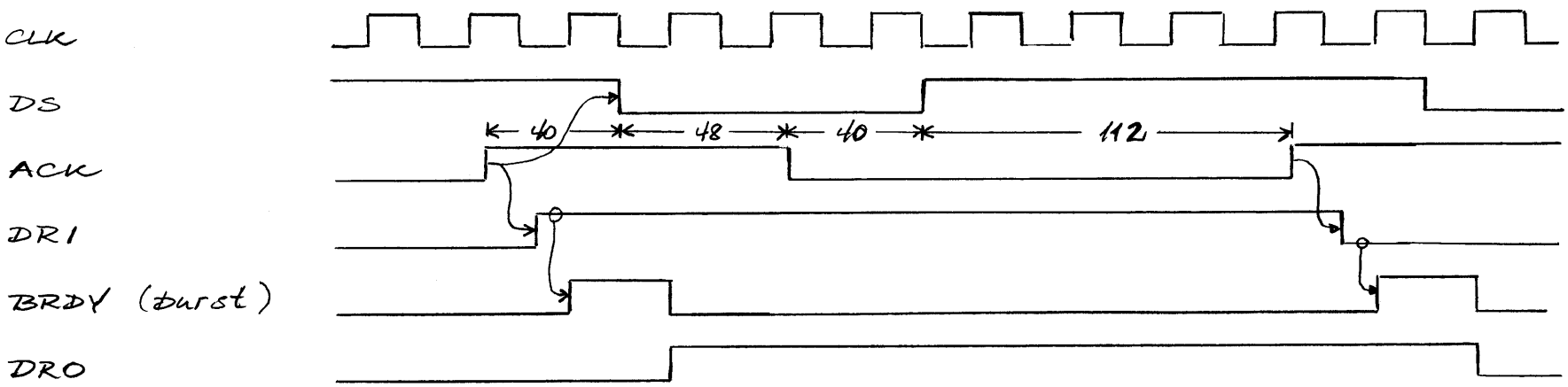
## Asynchronous State Machine for Read Cycles




## Asynchronous State Machine for Write Cycles



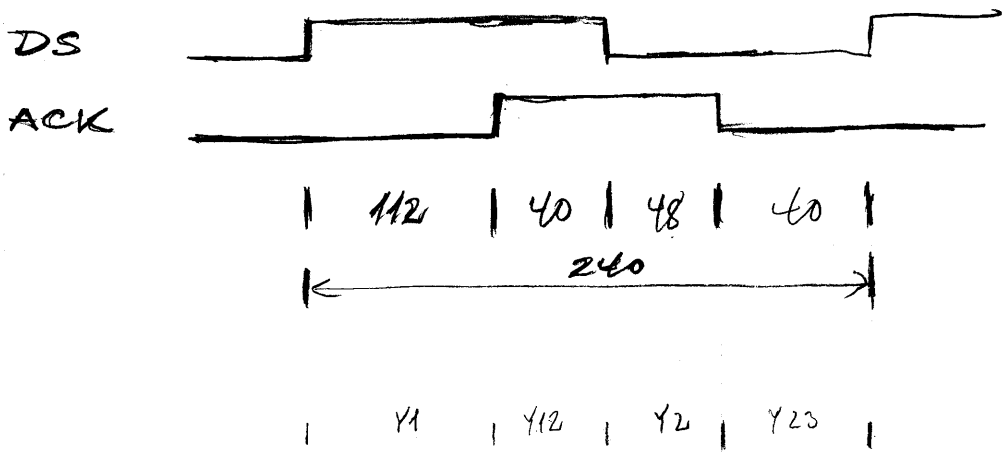
TPE301



AAJ / 96-10-09

TPE 301

TPE 200 + MACH 220-15



Dansk Data Elektronik A/S

| Control Overview | KAN | Page 1 |
|---|---|---|
| | 951106 | TPE301 |