

# In5Ctrl (file Oddu\_in) 10-5-2005\_16:00

## CMS CSC DDU5, Input Control FPGAs DF014A01 Version 14

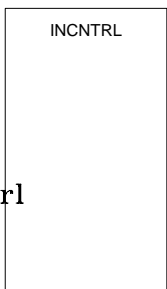
Process Rocket I/O Data from DMB, format output to DDU FIFOs v1: begin from in4ctrl v12  
 Last before DMB Format change--->> v2-9r2: v4: set FifoAlmostFull==HalfFullFifoA (un-invert),  
 require LFOK & FOK to Enable InUnit, Allow Start for NoLive Fibers case

this DDU: v10: invert Stat[31:28], add FakeL1en from VME, EvCntRst/BC0 to VMEctrl, DMBwarn to FMM from InUnit  
 RXER words are skipped, FILLER added as needed v11: tune MEM\_CTRL Search/Full logic  
 - Use code "C" with FILL flag set (b34 & b16) v12: fix FF0/1, retune MEM\_CTRL, add In\_C-CodeStat JTAG Reg (F20)  
 -- could use code "8" instead... v13: remove RxErr & NoLiveFibers from SingleErr, add RdCtrl-1 L1\_Count JTAG Reg (F26)  
 - Should we Reset RxErr Monitor? How? v14: tune Mon.Reg.Resets, add DMB Warn/Full monitor & register (f21), 2 I/O for IDMB\_Full to DDUctrl

### PART=XC2VP20-6-FG676

AVOID=Y21, E23, C22, E21  
 (INIT, BUSY, WRITE, CS)  
 NC\_XCV400\_FG676=B13, AF13  
 NC\_XCV400E\_FG676=D13, Y13

All I/O is 3.3V



#### Mode 1 Switch Block

- 1: Mode Bit 0
- 2: Mode Bit 1
- 3: Mode Bit 2
- 4: Mode Bit 3
- 7: Set Fake\_L1A (data passthrough)
- 8: Set all IO bits HIGH, ~FPGA version on LEDs

### PROM=2\*XC18V04-VQ44 (PARALLEL)

DDU5in\In5Ctrl\in5ctrl

PromID: 05026093h

FPGAid: 31266093h

#### VME Broadcast Addresses:

- 24=OSU-TCB "Test Control Board"
- 25=DMB
- 26=TMB
- 27=Both DMB and TMB
- 28=DDU
- 29=DCC

ELECTRONICS LAB  
 PHYSICS DEPARTMENT  
 THE OHIO STATE UNIVERSITY  
 174 WEST 18TH AVE  
 COLUMBUS OHIO 43210

- To Do:
- COMPARE L1NUM & BXN (DMB/TMB too)  
 add BX\_offset constants to SRAM?
  - use DDU-DMB fiber to STOP DMB on FULL, check
  - Watch for TRG buff overflows
  - No logic for TXEN, TXDIN, PAE0/1, Mode5/4...OK.

#### DDU Format Since DDUctrl v15:

H1: 0x/51/NN.NNNN/XXX/I.II/VK  
BOE type L1A Number BXN ID FOV K-Status

H2: 0x/8000/0001/8000/HHHH  
49-bit-unique constant DMBull(15)

H3: 0x/LLLL/oooo/ZZZZ/GMY  
LiveDMB(15) Dstat DMB-DW(15) BOEstat DMBent

T-2: 0x/8000/FFFF/8000/8000  
64-bit unique constant

T-1: 0x/SSSS.SSSS/QQQQ/PPPP  
DDU EOE Status DMBerr DMBwarn(15)

TR: 0x/A/?WWW.WWWW/RRRR/UUMK  
EOE WordCount PRGword EOEstat K-Status

DDU WordCount (64-bit words) for "No Data" event: 0x006.  
 DDU WordCount for one DMB (only one CFEB): 0D2h = 210 dec, 1680 Bytes  
 DDU WC, 1 DMB with 2 CFEB (8 samples each): 19Ah = 410 dec, 3280 Bytes  
 DDU WC, 2 DMB with 1 CFEB (nCFEB=2): 19Eh = 414 dec, 3312 Bytes  
 DDU WC, 2 DMB with 2 CFEB (nCFEB=4): 32Eh = 814 dec, 6512 Bytes

DDU WordCount = (6 + 25\*Nts\*nCFEB + 4\*nDMB) < 30070; 240560 Bytes  
 ^Ignores TMB Data^ GBF ByteCount = 8\*DDU WordCount 8 TS assumed

DDU WC, 3 DMB with 1 CFEB (nCFEB=3): 26Ah = 618 dec, 4944 Bytes  
 DDU WC, 4 DMB with 1 CFEB (nCFEB=4): 336h = 822 dec, 6576 Bytes  
 DDU WC, 7 DMB with 1 CFEB (nCFEB=7): 59Ah = 1434 dec, 11472 Bytes  
 DDU WC, 8 DMB with 1 CFEB (nCFEB=8): 666h = 1638 dec, 13104 Bytes

DDU WC, 11 DMB with 1 CFEB (nCFEB=11): 8CAh = 2250 dec, 18000 Bytes  
 DDU WC, 12 DMB with 1 CFEB (nCFEB=12): 996h = 2454 dec, 19632 Bytes  
 DDU WC, 15 DMB with 1 CFEB (nCFEB=15): BFAh = 3066 dec, 24528 Bytes

Mask DMBs with critical error until they're Reset?

--->Not useful, comes 3 words too late to stop data

Single error, insignificant unless repeated

Tied to GND. Add DMB status checks.

Remove? <<<<

This event is garbage! Set error bit in SLINK...

SCA\_OVFL--->>

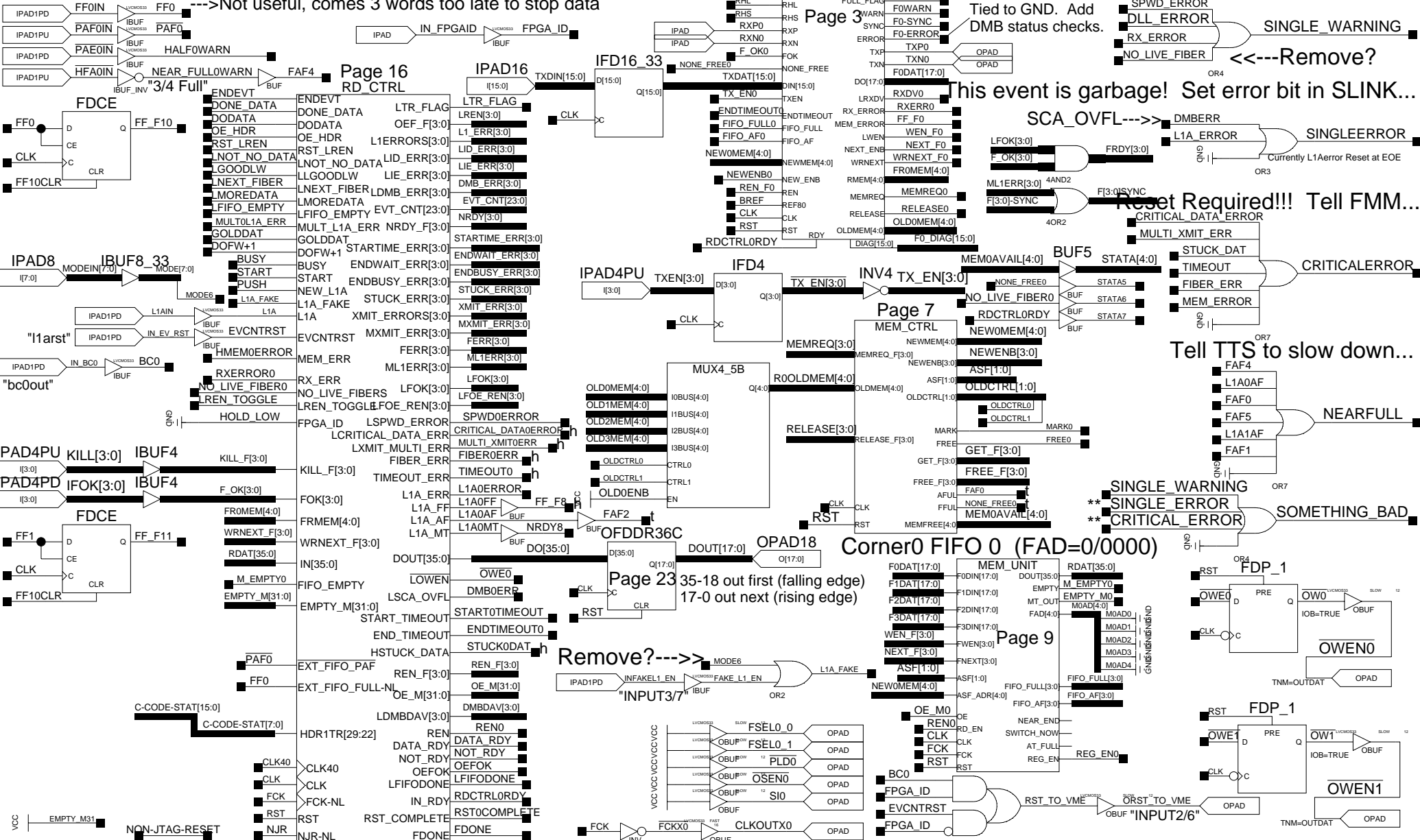
Reset Required!!! Tell FMM...

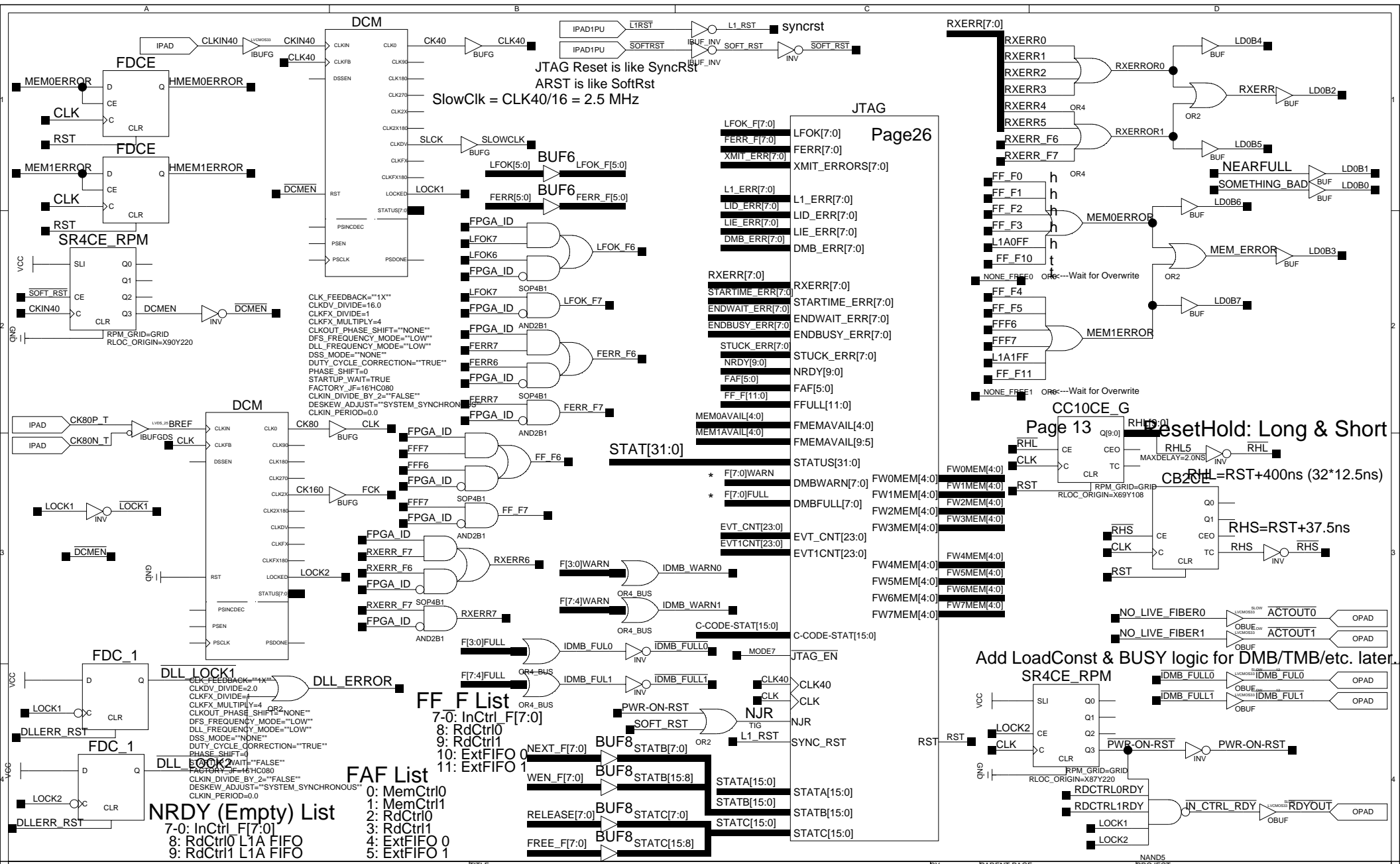
Tell TTS to slow down...

Corner0 FIFO 0 (FAD=0/0000)

35-18 out first (falling edge)  
17-0 out next (rising edge)

Remove?--->>





JTAG Reset is like SyncRst  
 ARST is like SoftRst  
 SlowClk = CLK40/16 = 2.5 MHz

Page 26

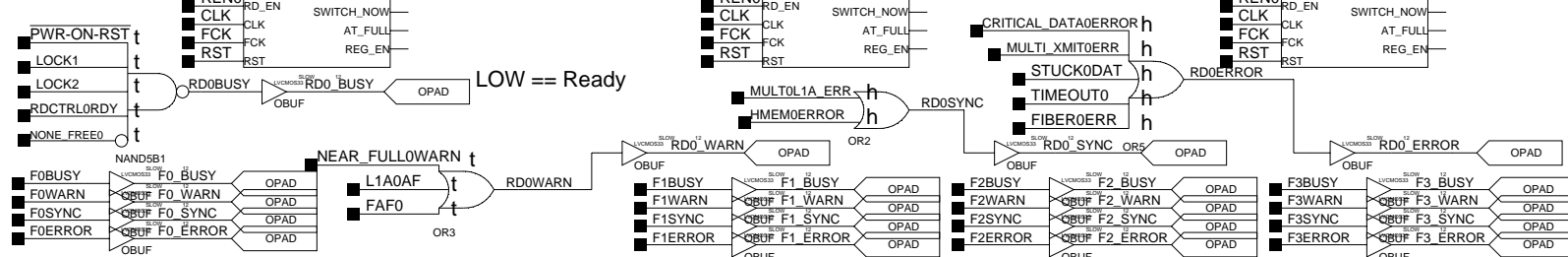
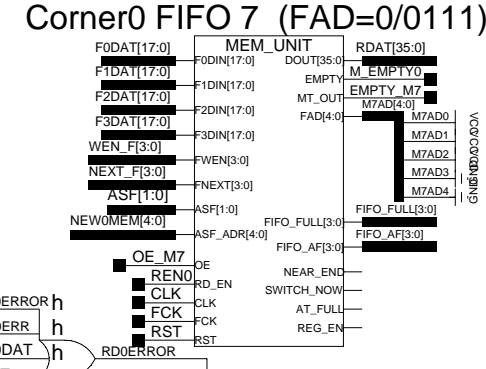
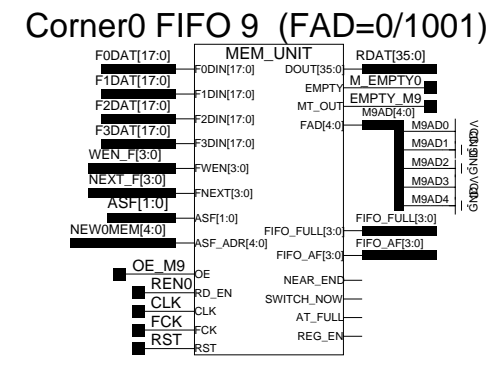
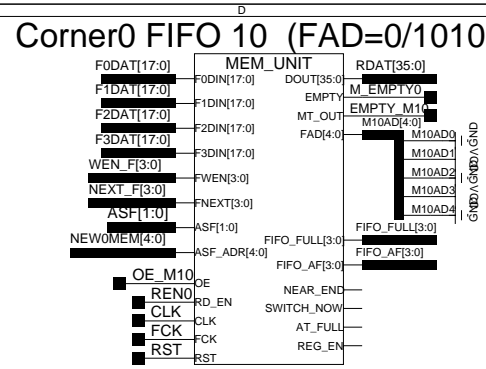
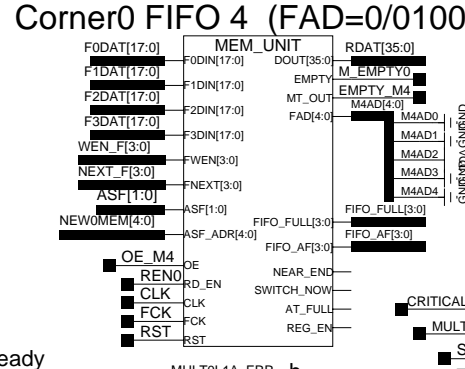
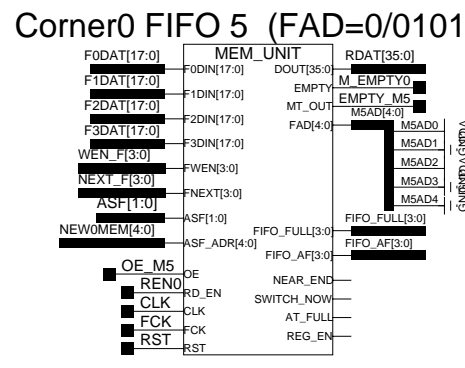
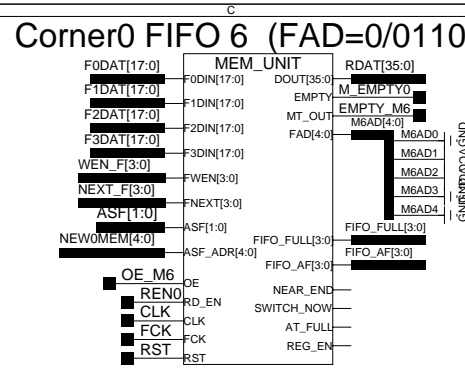
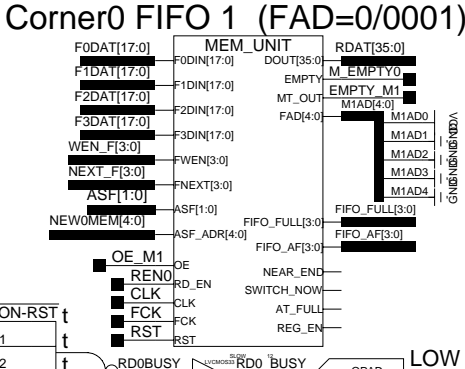
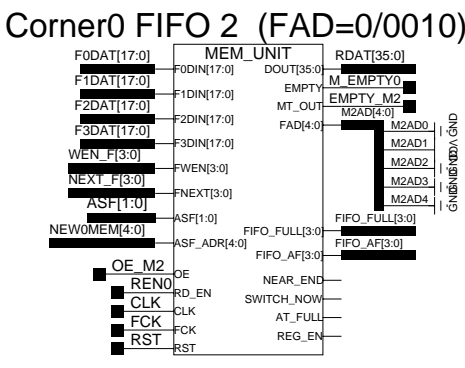
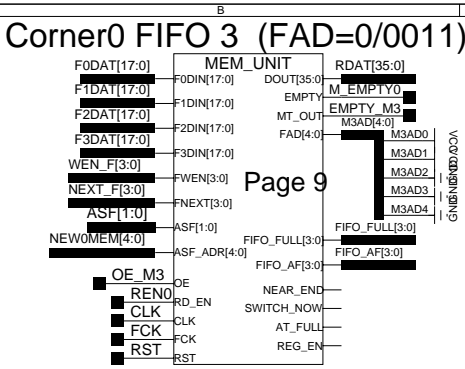
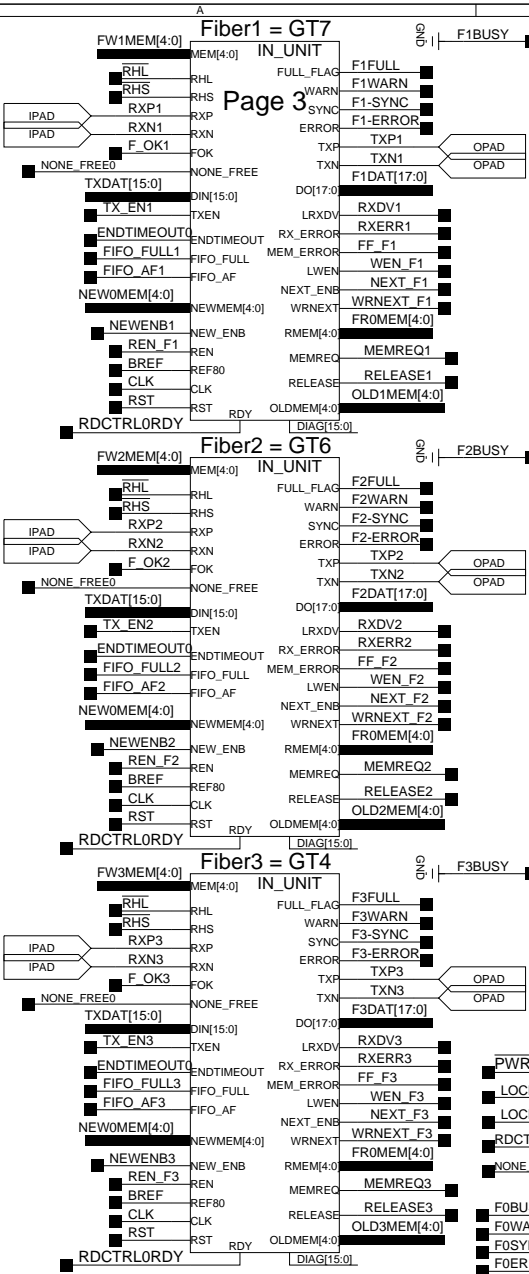
Page 13 ResetHold: Long & Short

Add LoadConst & BUSY logic for DMB/TMB/etc. later.

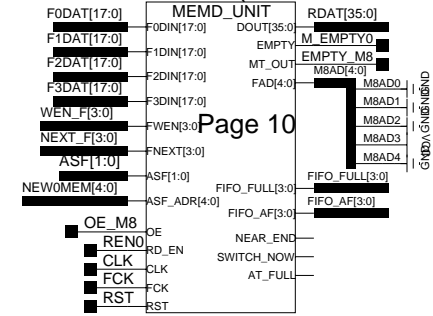
FF F List  
 7-0: InCtrl\_F[7:0]  
 8: RdCtrl0  
 9: RdCtrl1  
 10: ExtFIFO 0  
 11: ExtFIFO 1

FAF List  
 0: MemCtrl0  
 1: MemCtrl1  
 2: RdCtrl0  
 3: RdCtrl1  
 4: ExtFIFO 0  
 5: ExtFIFO 1

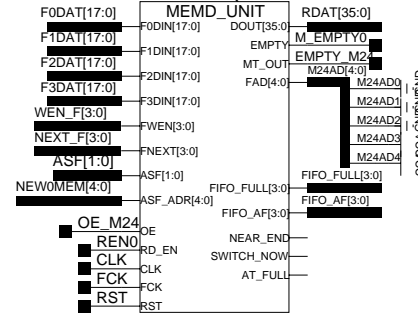
NRDY (Empty) List  
 7-0: InCtrl\_F[7:0]  
 8: RdCtrl0 L1A FIFO  
 9: RdCtrl1 L1A FIFO



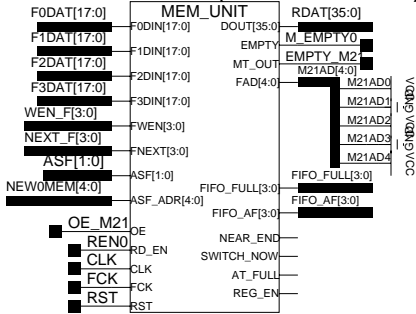
Corner0 FIFO 8\* (FAD=0/1000)



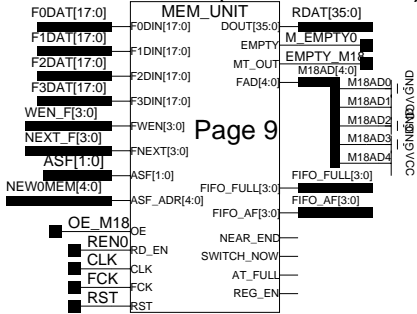
Corner1 FIFO 8\* (FAD=1/1000)



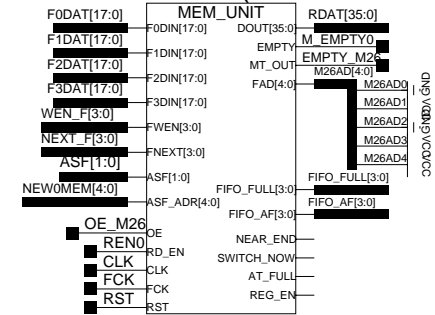
Corner1 FIFO 5 (FAD=1/0101)



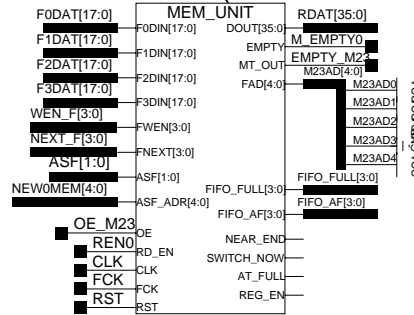
Corner1 FIFO 2 (FAD=1/0010)



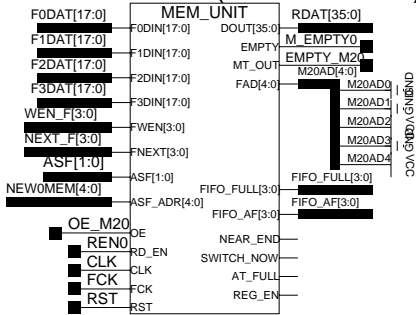
Corner1 FIFO 10 (FAD=1/1010)



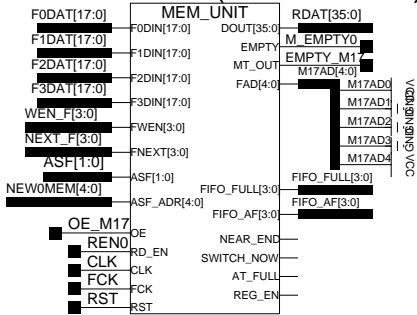
Corner1 FIFO 7 (FAD=1/0111)



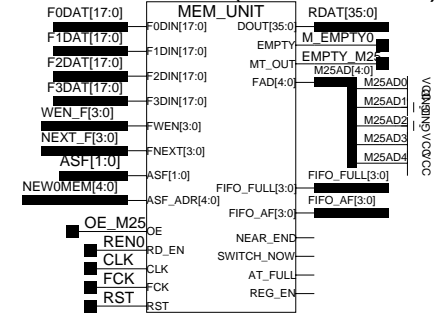
Corner1 FIFO 4 (FAD=1/0100)



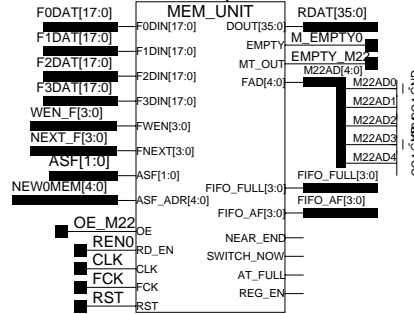
Corner1 FIFO 1 (FAD=1/0001)



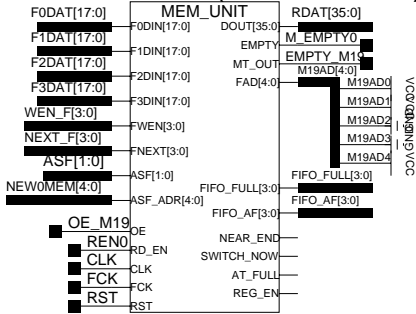
Corner1 FIFO 9 (FAD=1/1001)



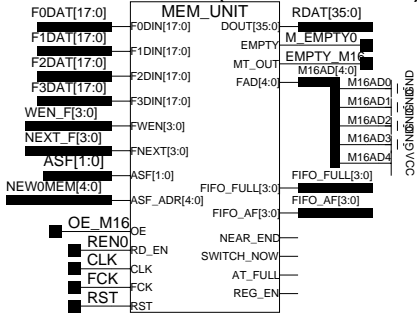
Corner1 FIFO 6 (FAD=1/0110)

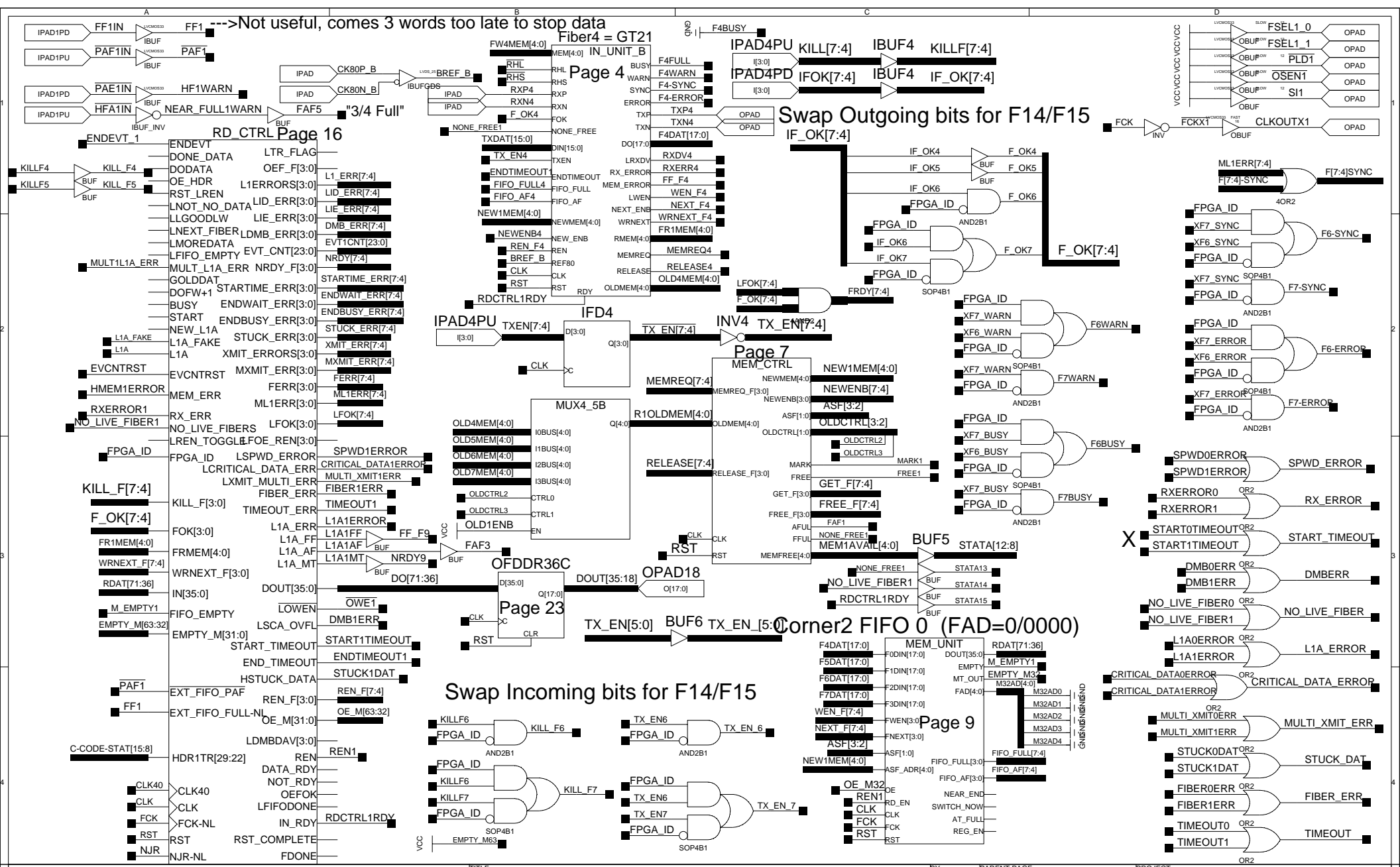


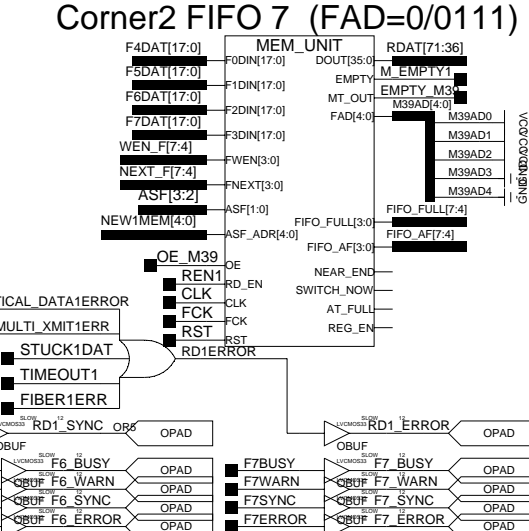
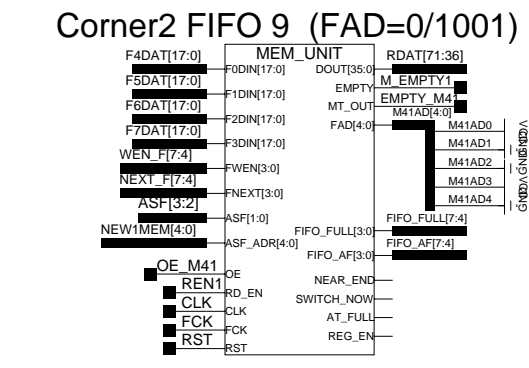
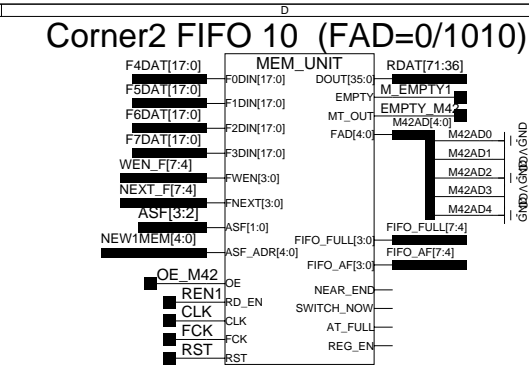
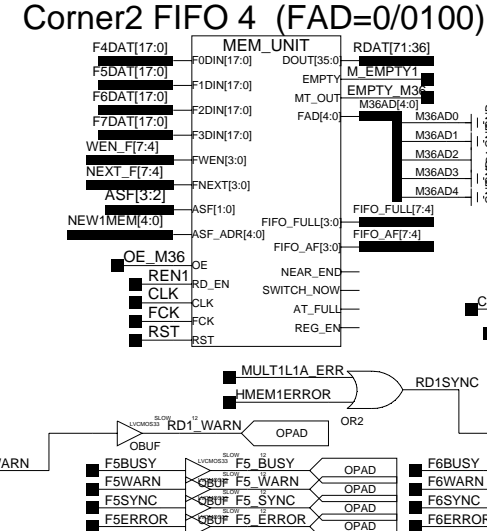
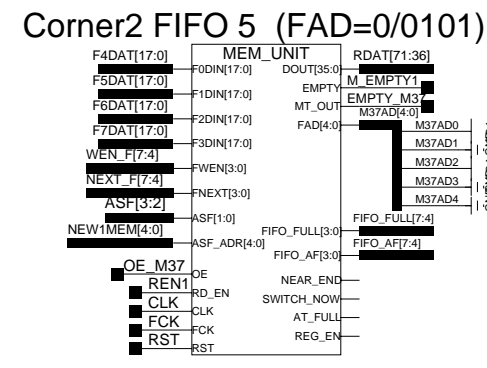
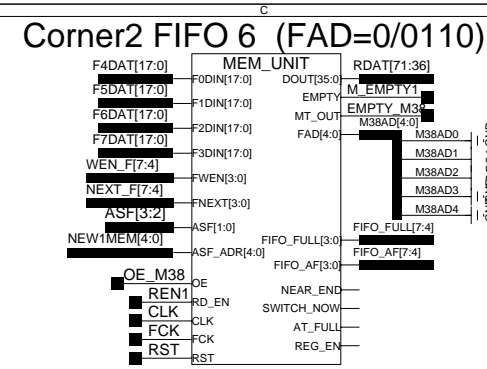
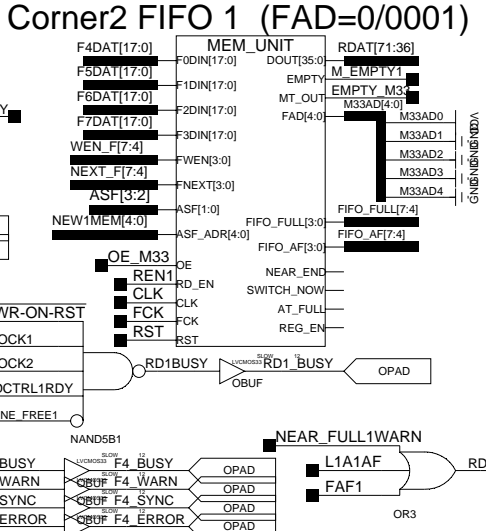
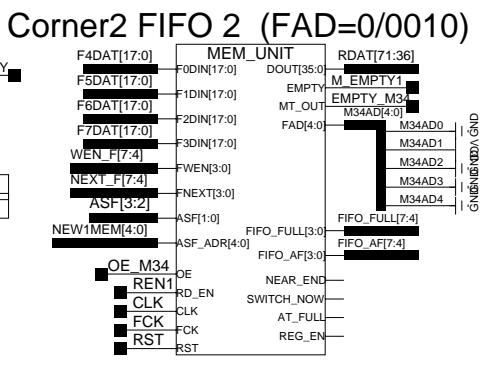
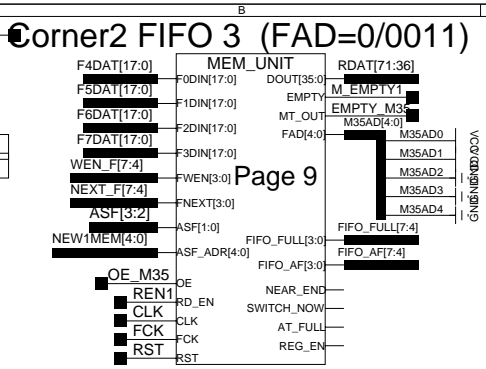
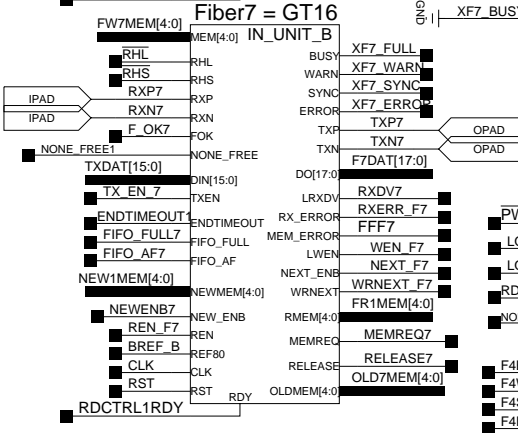
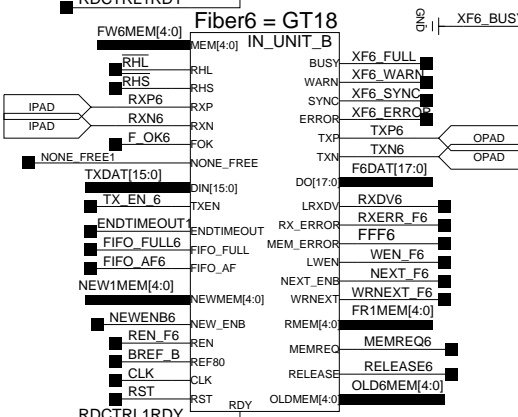
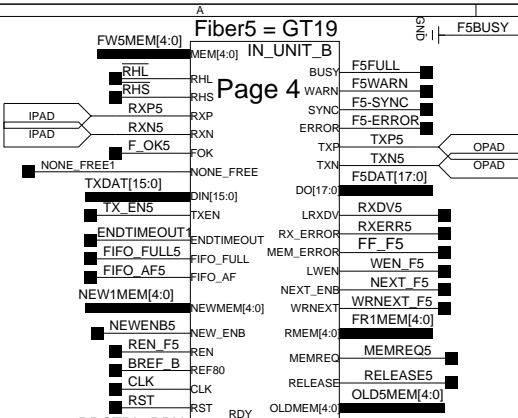
Corner1 FIFO 3 (FAD=1/0011)



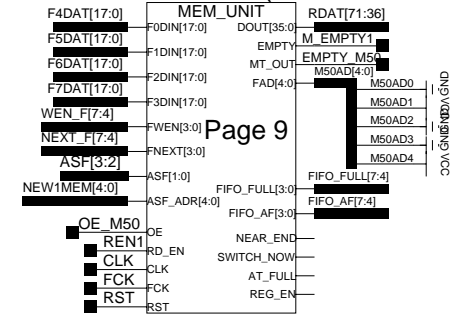
Corner1 FIFO 0 (FAD=1/0000)





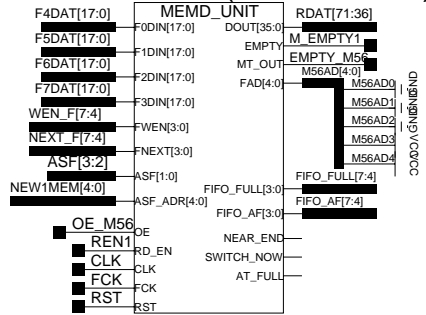
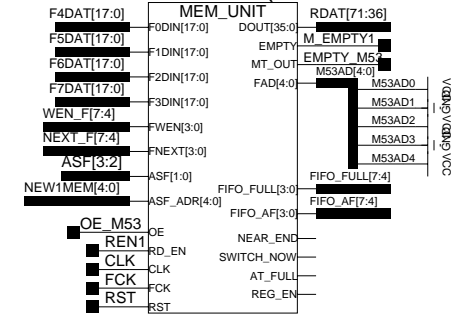


Corner3 FIFO 2 (FAD=1/0010)

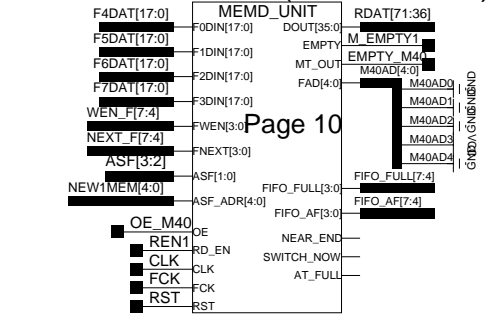


Page 9

Corner3 FIFO 5 (FAD=1/0101) Corner3 FIFO 8\* (FAD=1/1000)

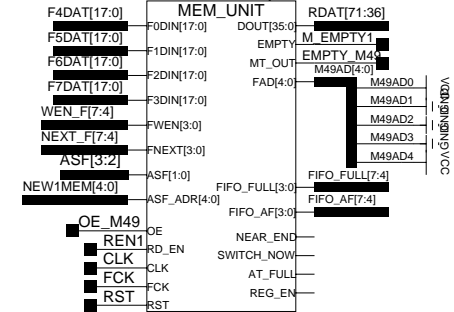


Corner2 FIFO 8\* (FAD=1/1000)

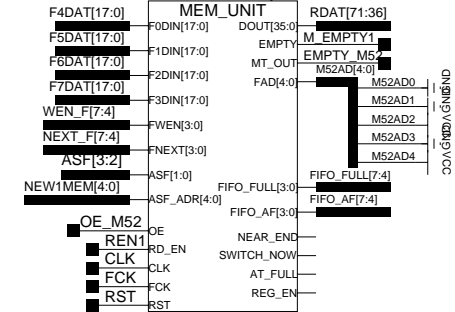


Page 10

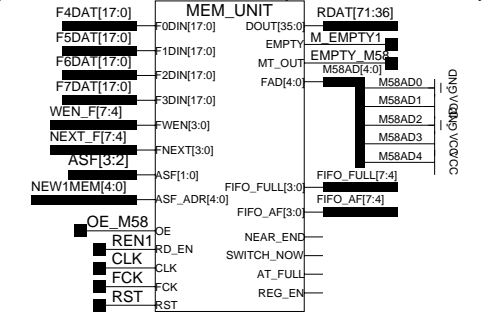
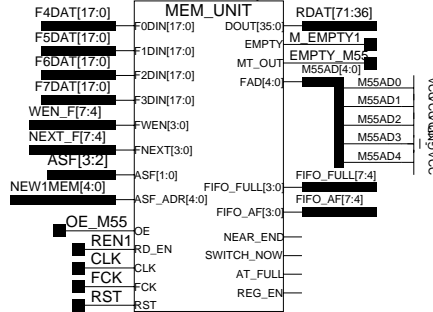
Corner3 FIFO 1 (FAD=1/0001)



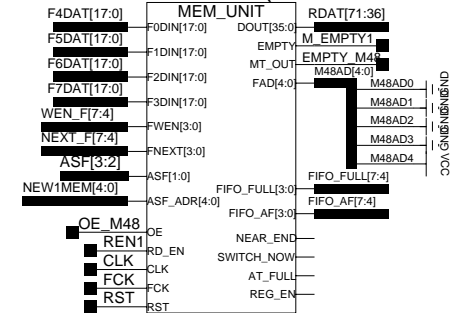
Corner3 FIFO 4 (FAD=1/0100)



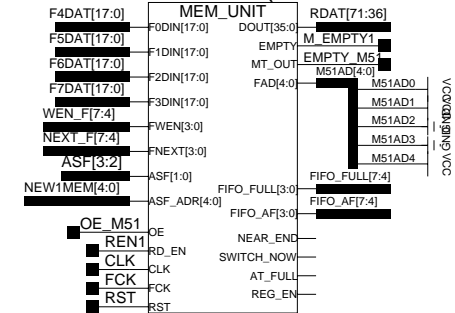
Corner3 FIFO 7 (FAD=1/0111) Corner3 FIFO 10 (FAD=1/1010)



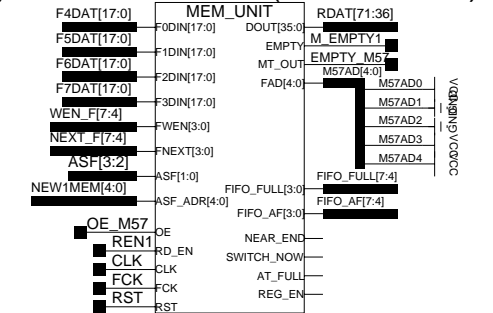
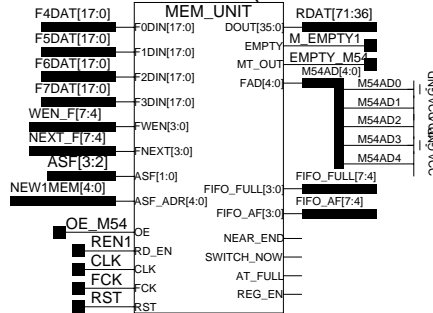
Corner3 FIFO 0 (FAD=1/0000)



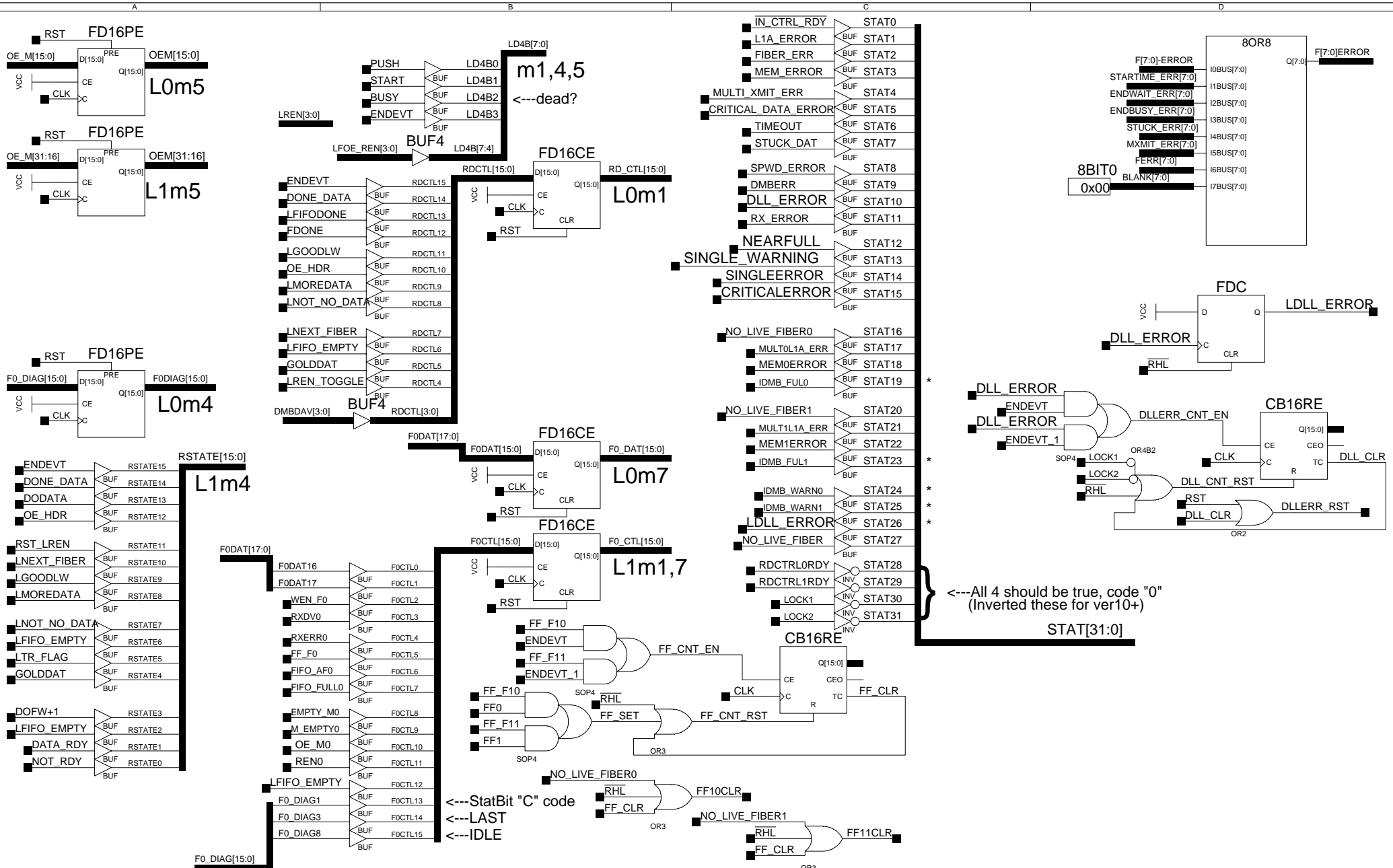
Corner3 FIFO 3 (FAD=1/0011)



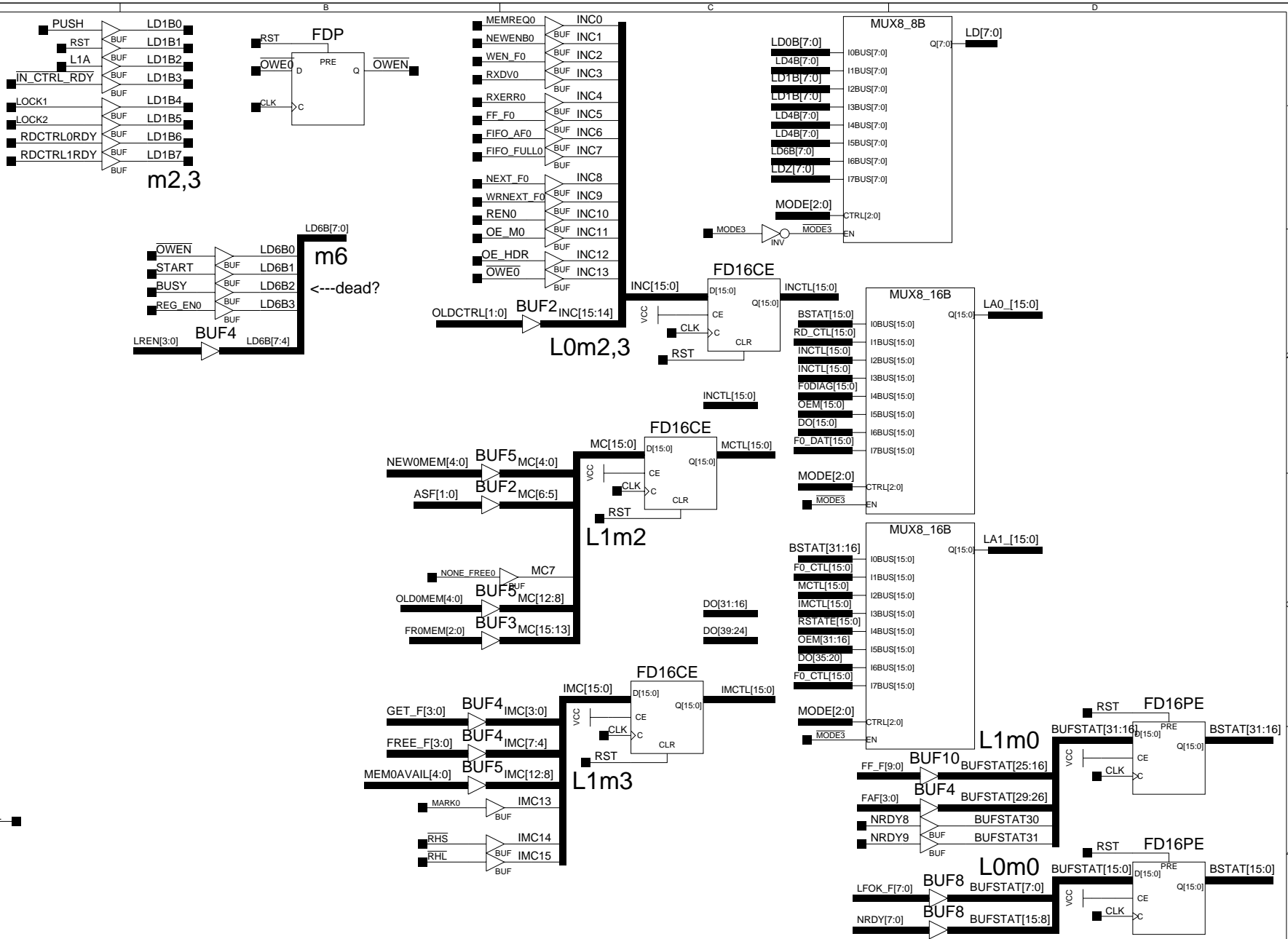
Corner3 FIFO 6 (FAD=1/0110) Corner3 FIFO 9 (FAD=1/1001)





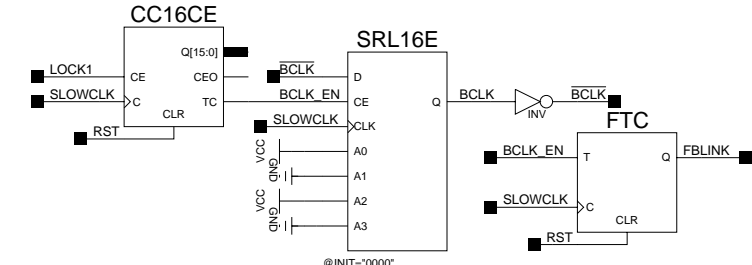
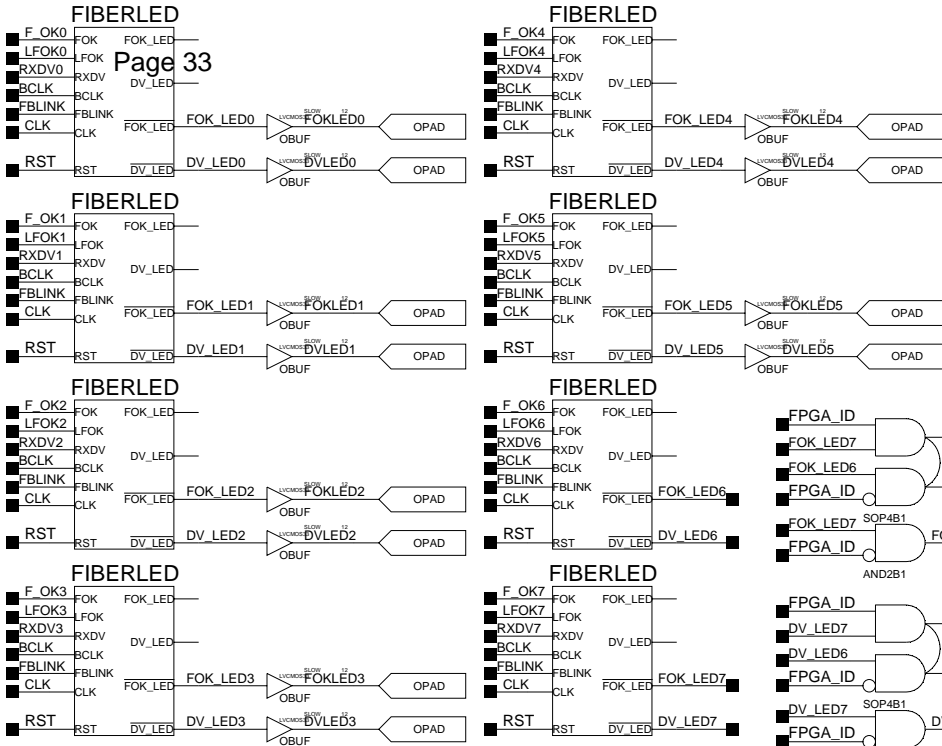


<---All 4 should be true, code "0"  
(Inverted these for ver10+)

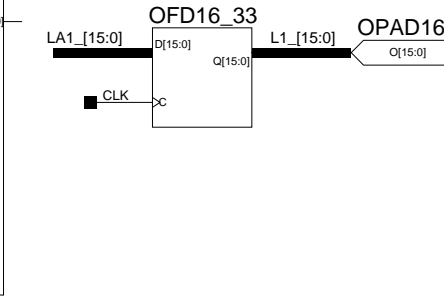
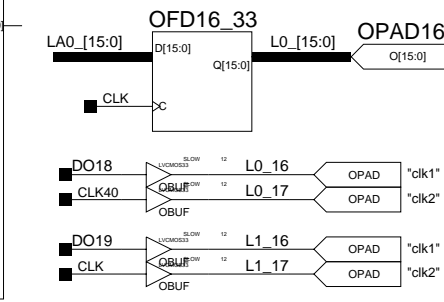
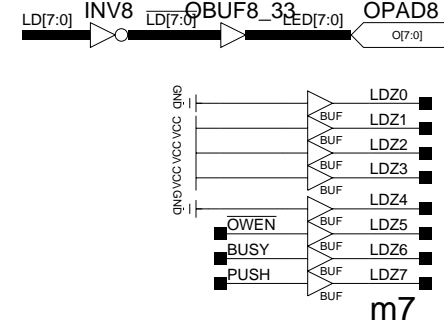
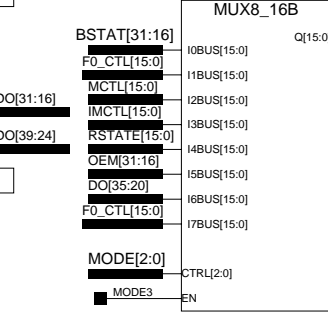
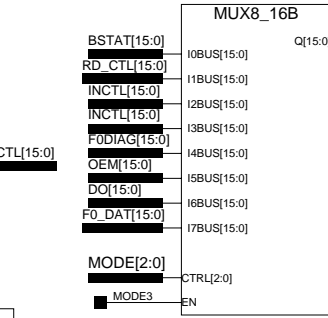
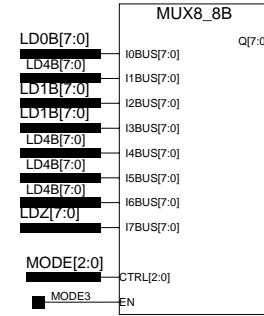


# END

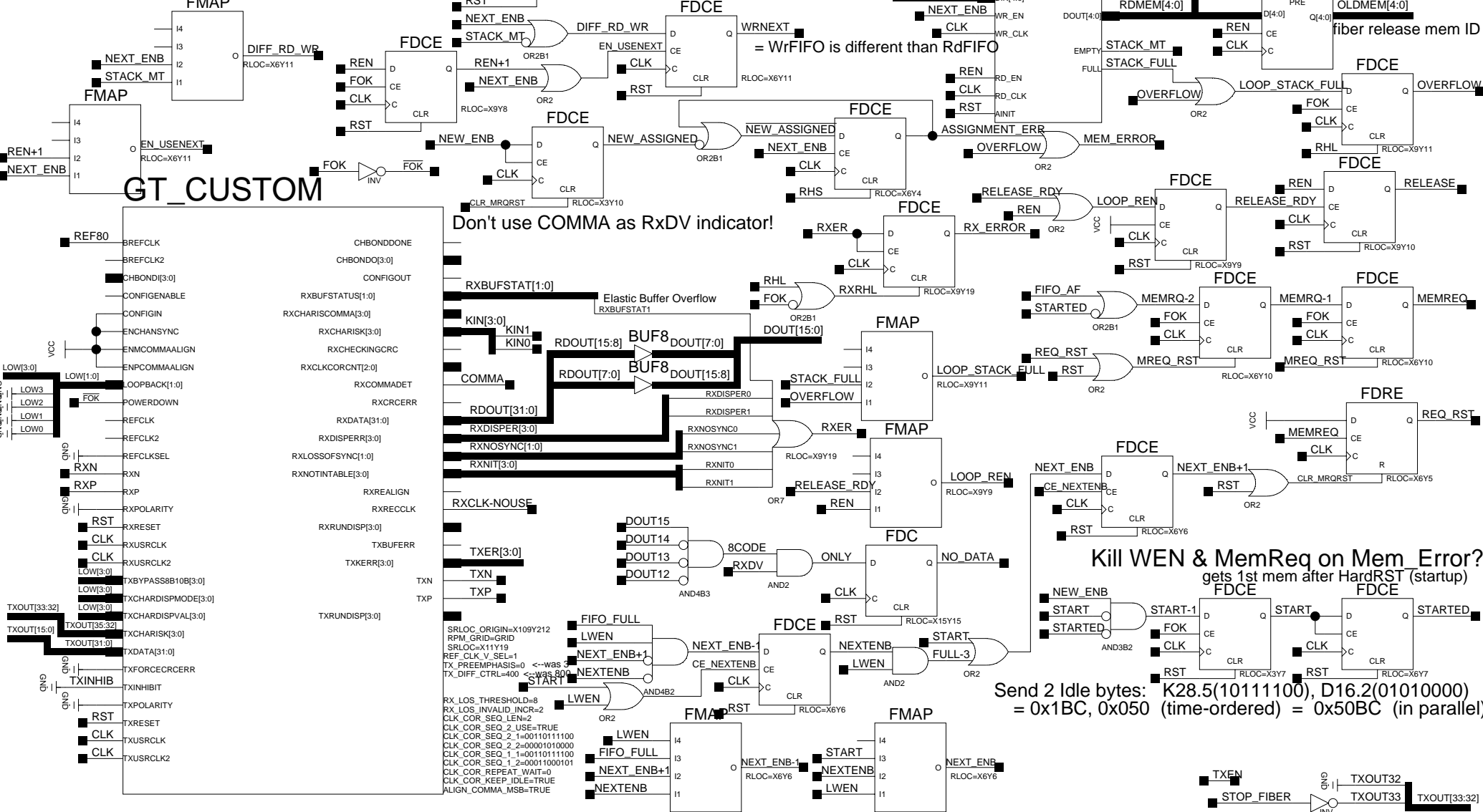
Page 33



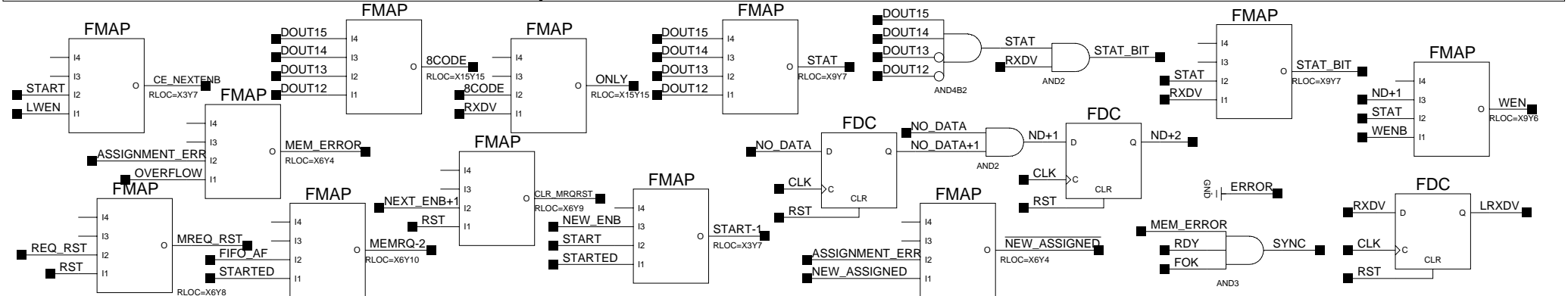
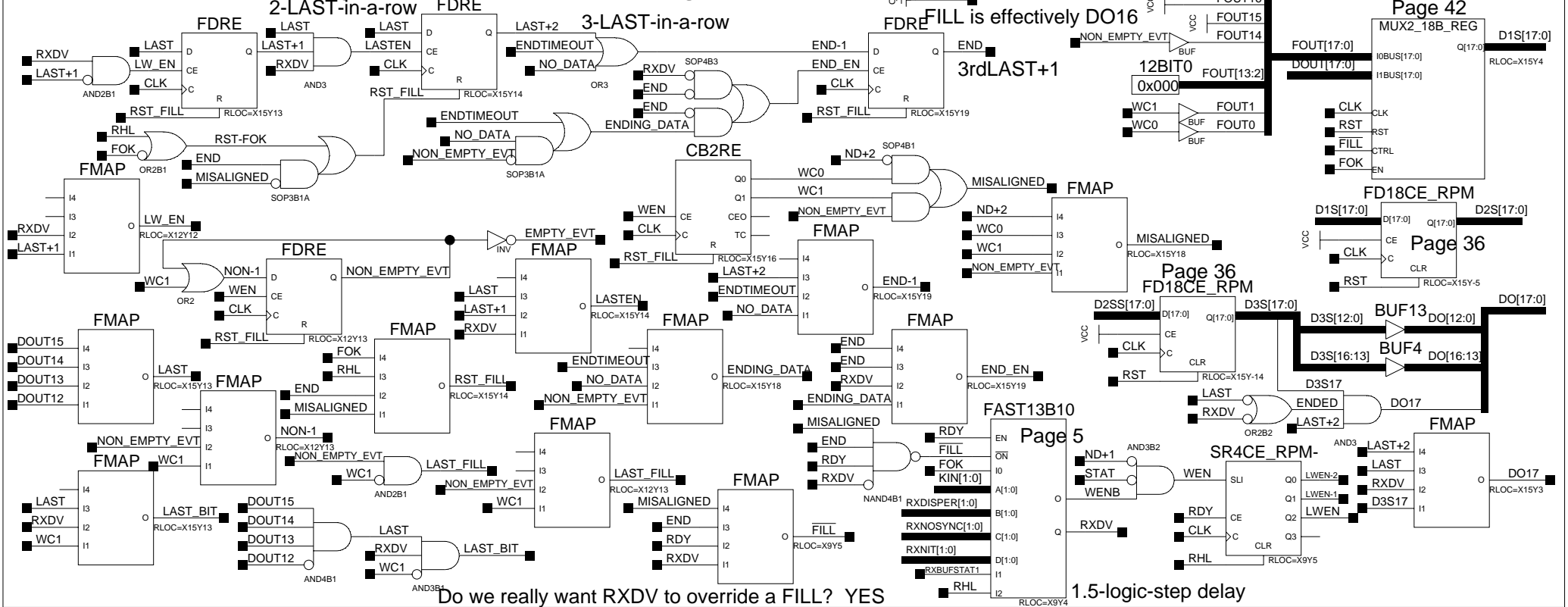
SlowCLK=2.5MHz, BCLK\_EN=38 Hz. Shift 13 will give ~1.5Hz BCLK, Shift 5 gives ~4Hz.

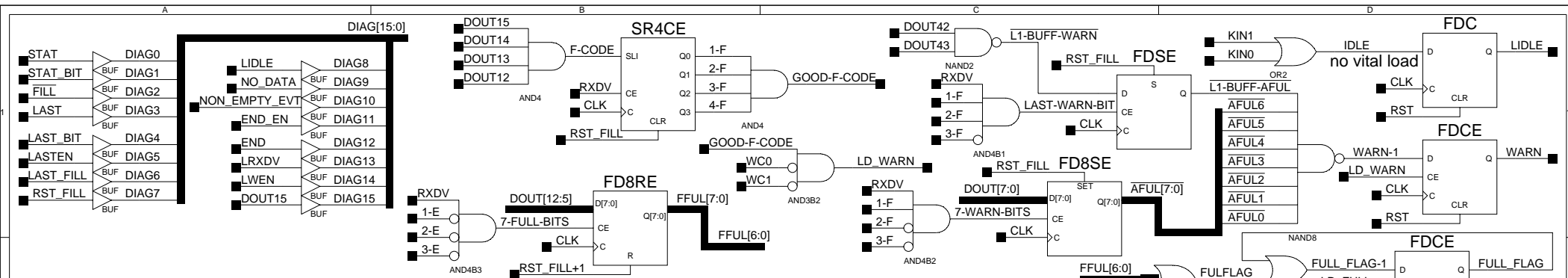


FIFO\_AF & FIFO\_FULL are driven by Tbufs from 22 mem's.  
 NEW\_ENB & NEW\_MEM are driven by MemCtrl.  
 TX\_EN & DIN are driven by DDU\_Ctrl FPGA.  
 REN is driven by RD\_CTRL.  
 IDLEOUT needs local control logic.



# FILL Logic for 2-4 word alignment





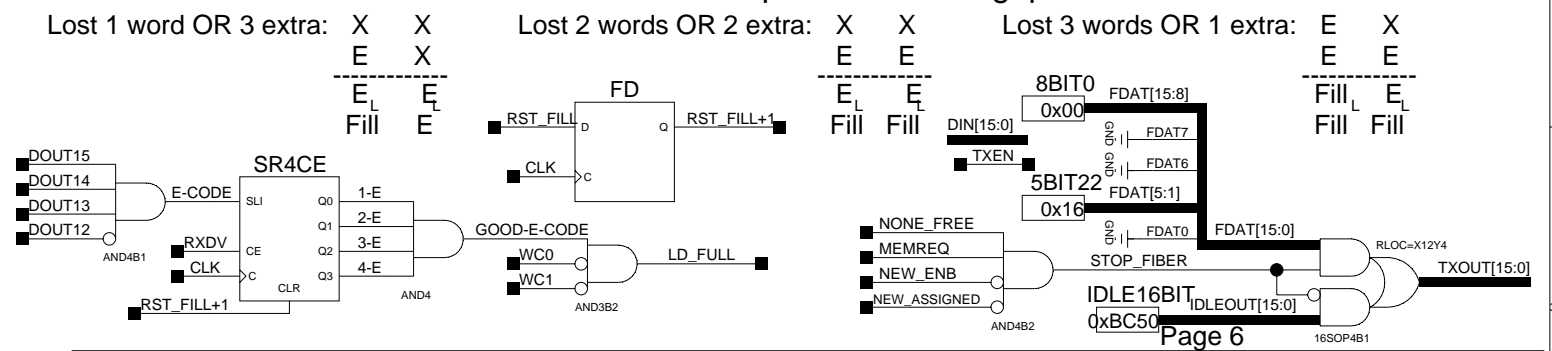
LAST Flag is insensitive to single-bit errors

AST Flag Logic tables, checks for E-Codes in DMB Data (L means LAST Flag will be set for that word)

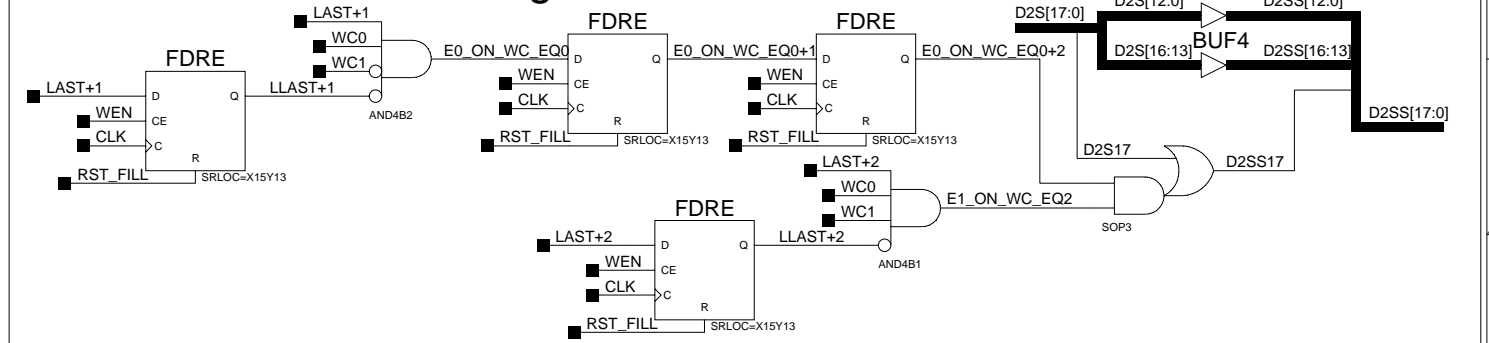
BRAM FIFO Output

\*Assume 4 E-Codes at DMB End are sequential with no gaps\*

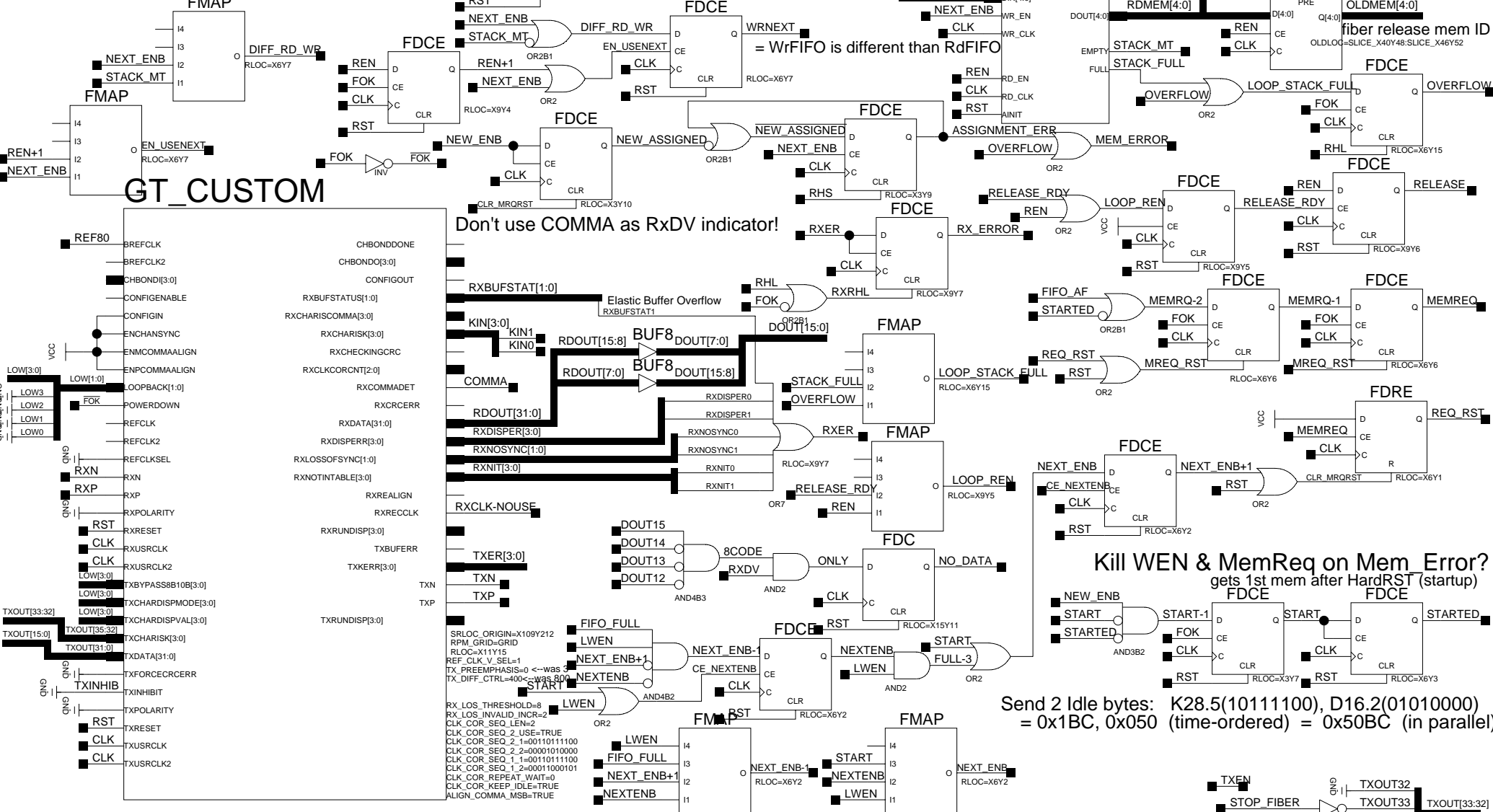
Normal event sync:	X	X
	X	X
	-----	
	E <sub>L</sub>	E
	E	E
Normal, but lose 1st E-Word: (or lose last E-Word)	X	X
	X	X
	-----	
	E <sub>L</sub>	E
	Fill	E
Normal, but lose 2nd E-Word: (or lose 3rd E-Word)	X	X
	X	X
	-----	
	E <sub>L</sub>	E
	Fill	E
Normal, but BAD 1st E-Code:	X	X
	X	X
	-----	
	E <sub>L</sub>	X
	E	E
Normal, but BAD 2nd E-Code:	X	X
	X	X
	-----	
	X <sub>L</sub>	E
	E	E
BAD 3rd or 4th E-Code is no problem		



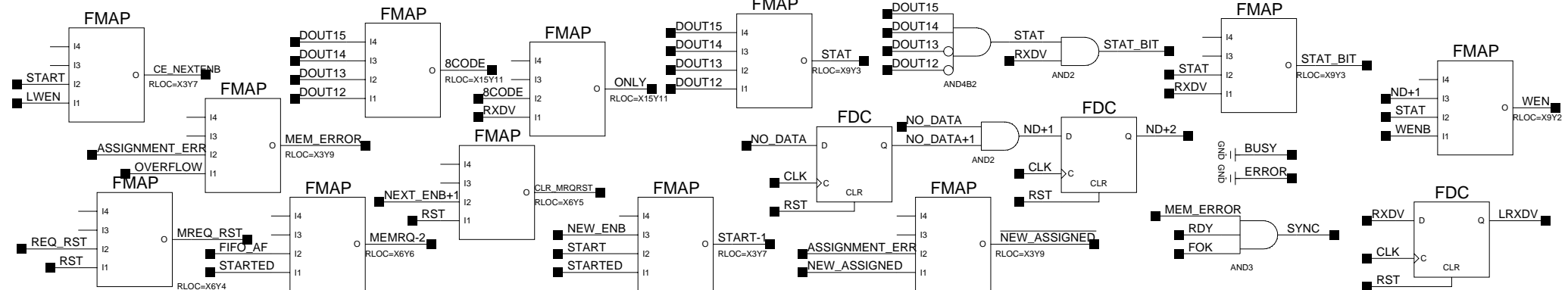
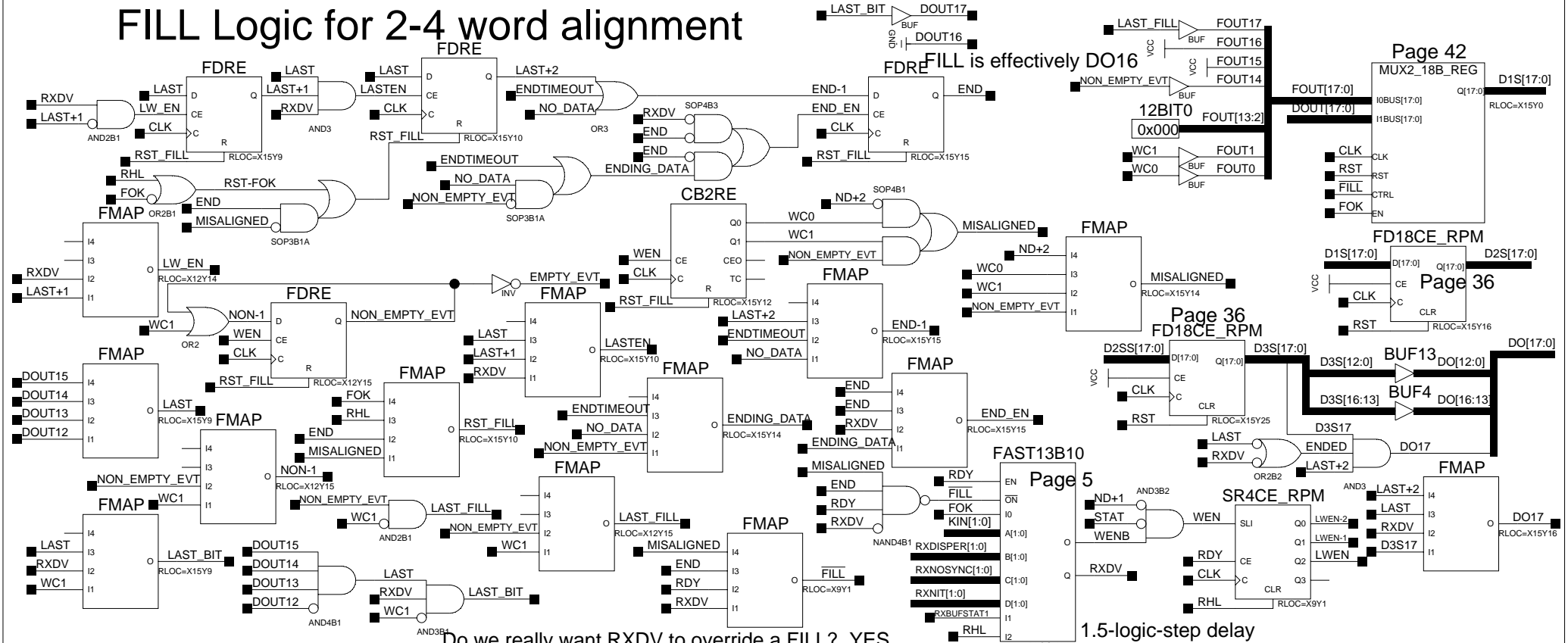
### Fix to set LAST Flag for Bad 2nd E-Code case



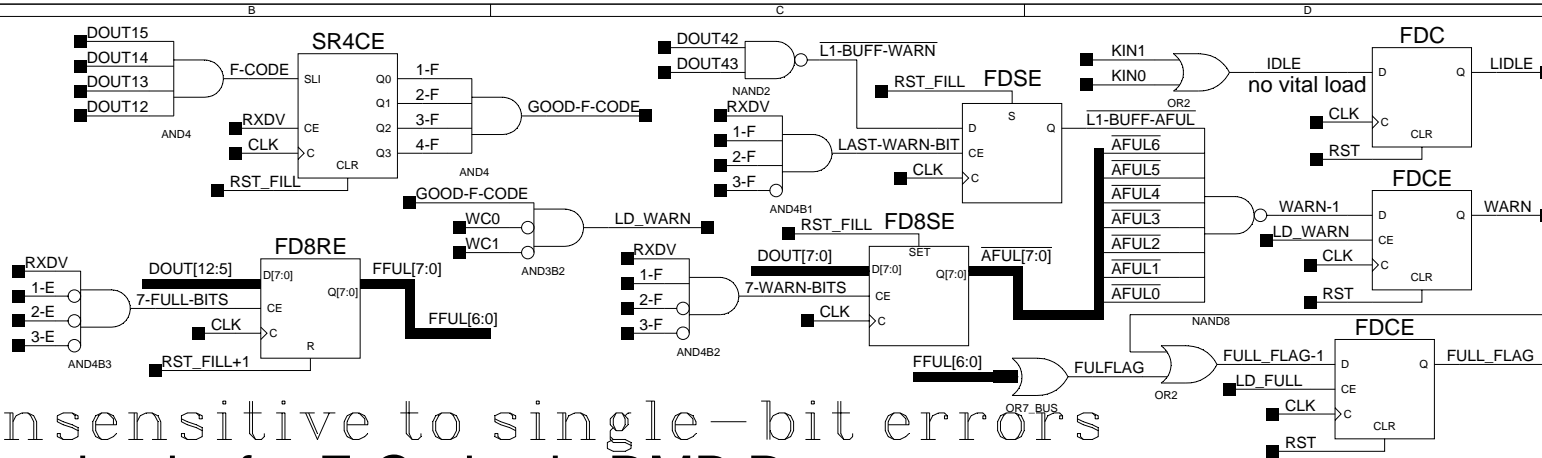
FIFO\_AF & FIFO\_FULL are driven by Tbufs from 22 mem's.  
 NEW\_ENB & NEW\_MEM are driven by MemCtrl.  
 TX\_EN & DIN are driven by DDU\_Ctrl FPGA.  
 REN is driven by RD\_CTRL.  
 IDLEOUT needs local control logic.



# FILL Logic for 2-4 word alignment







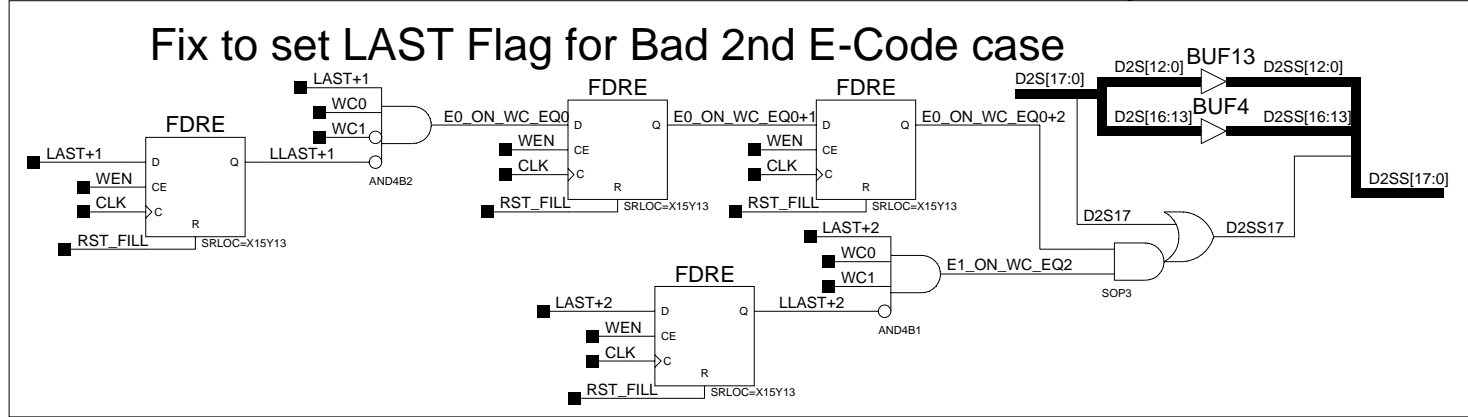
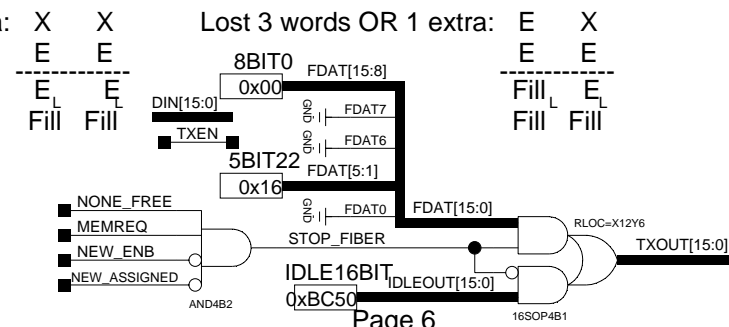
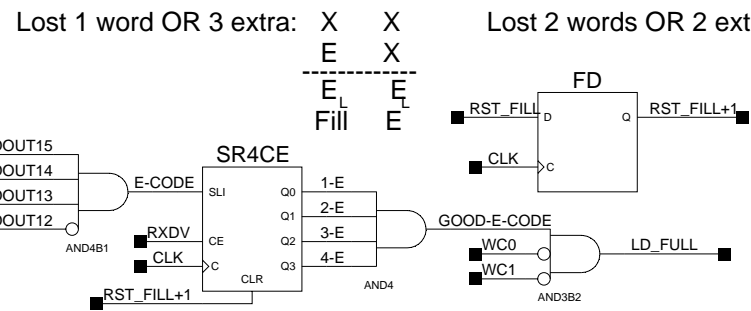
LAST Flag is insensitive to single-bit errors

LAST Flag Logic tables, checks for E-Codes in DMB Data (L means LAST Flag will be set for that word)

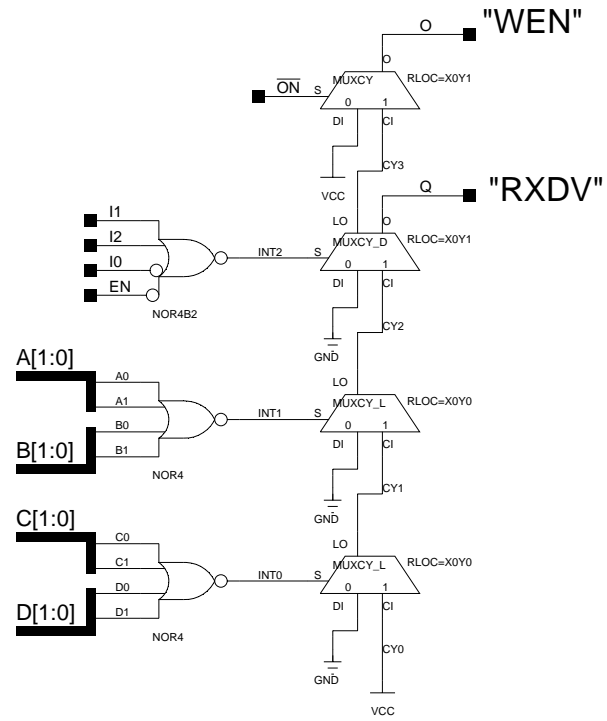
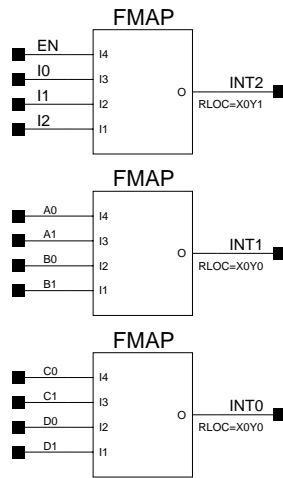
BRAM FIFO Output

\*Assume 4 E-Codes at DMB End are sequential with no gaps\*

Normal event sync:	X	X
	X	X
<hr/>		
	E <sub>L</sub>	E
	E	E
Normal, but lose 1st E-Word: (or lose last E-Word)	X	X
	X	X
<hr/>		
	E <sub>L</sub>	E <sub>L</sub>
	Fill	E
Normal, but lose 2nd E-Word: (or lose 3rd E-Word)	X	X
	X	X
<hr/>		
	E <sub>L</sub>	E
	Fill	E
Normal, but BAD 1st E-Code:	X	X
	X	X
<hr/>		
	E <sub>L</sub>	X
	E	E
Normal, but BAD 2nd E-Code:	X	X
	X	X
<hr/>		
	X <sub>L</sub>	E
	E	E
BAD 3rd or 4th E-Code is no problem		



Page 6

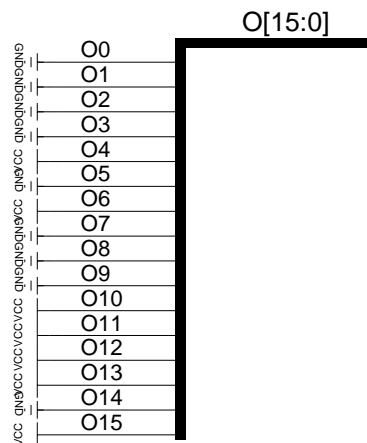


JRG

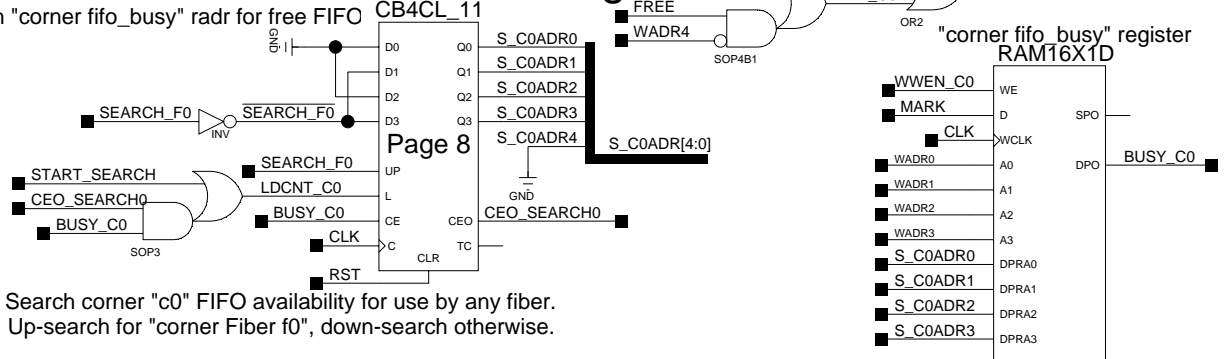
Title:	FAST13B10	
Comments:	Custom Logic for DDU similar to: AND12B10 with an OR2 (allows ON to override)	
Date:	19th December 2003	Ver: 1
Sheet Size:	B	Rev: A

Send 2 Idle bytes:

$K28.5(10111100)+D16.2(01010000)$   
= 0x1BC + 0x050 (time-ordered)  
= 0xBC50 (in parallel)



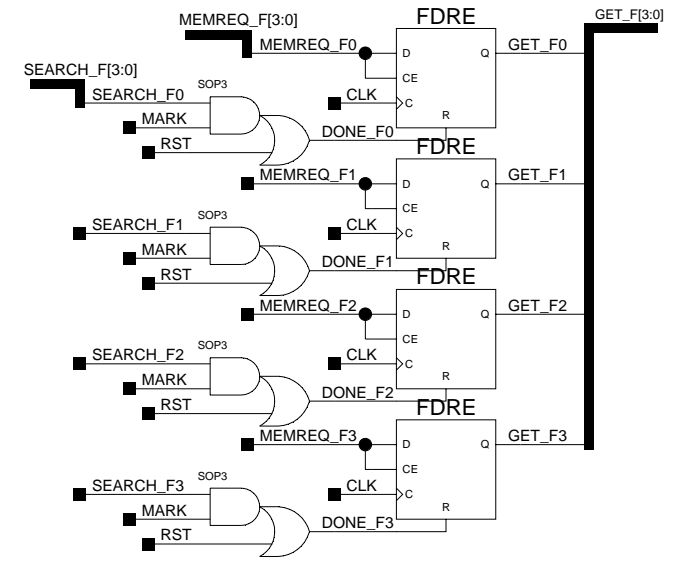
# Corner 0 FIFO usage



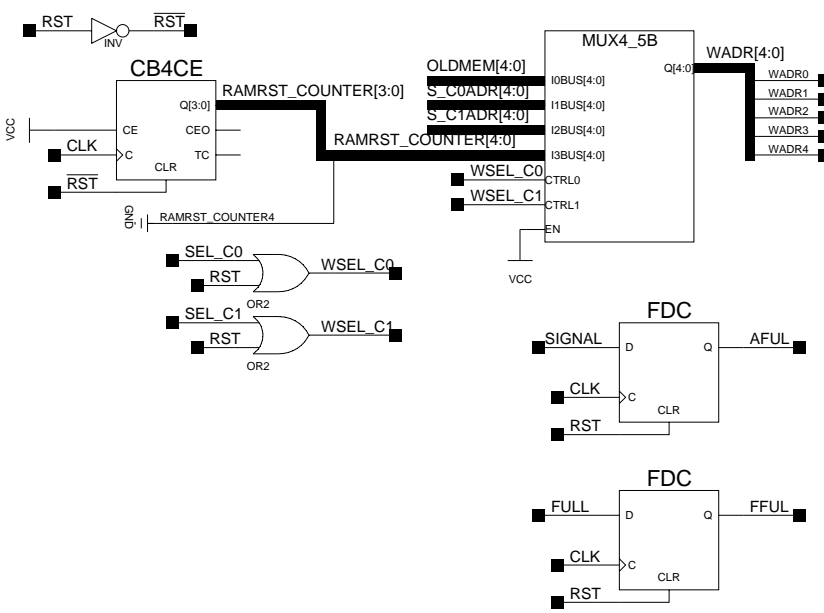
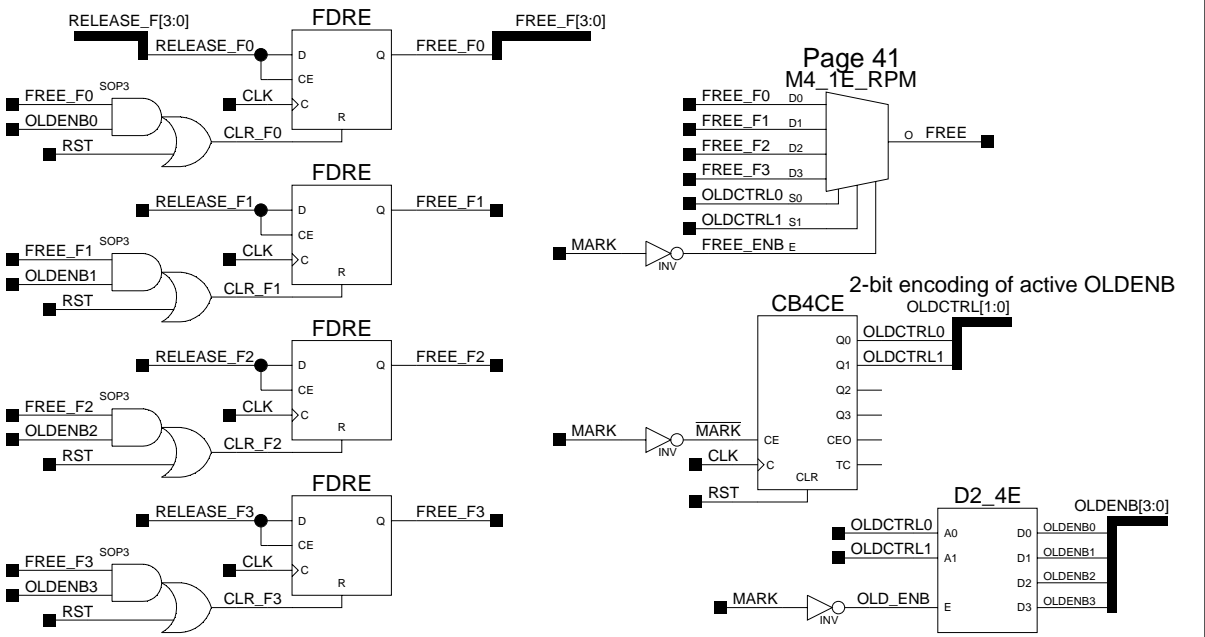
Search corner "c0" FIFO availability for use by any fiber.  
Up-search for "corner Fiber f0", down-search otherwise.

write "0" into adr "oldmem" to mark FIFO available

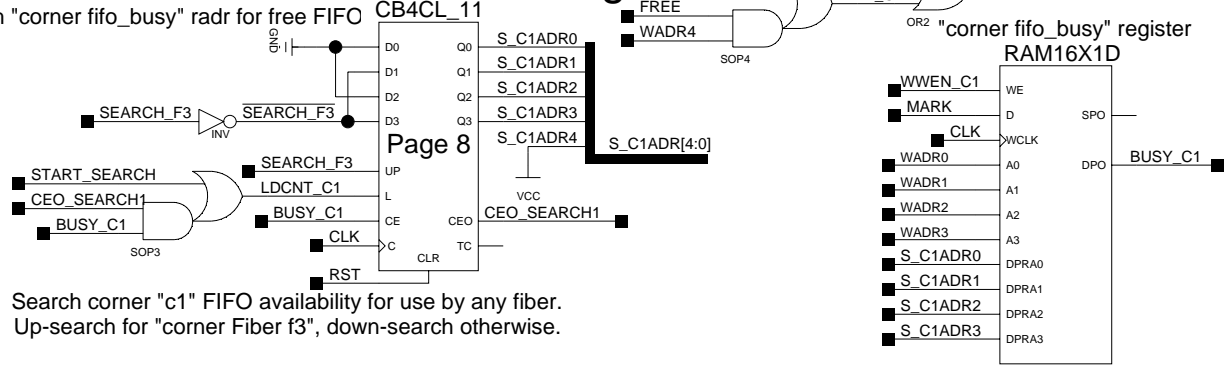
# Which Fibers need a new FIFO?



# FIFO Release Control



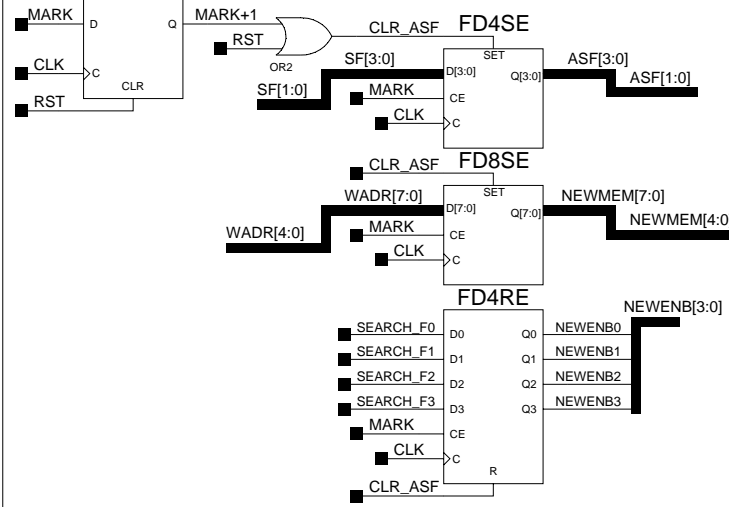
# Corner 1 FIFO usage



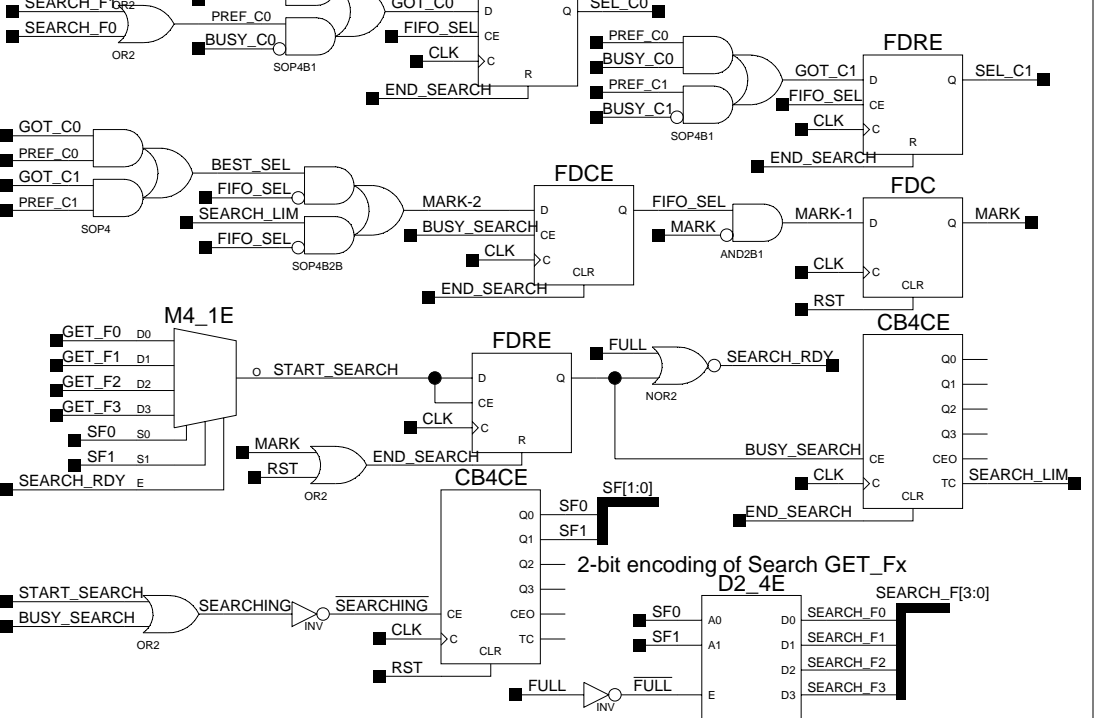
Search corner "c1" FIFO availability for use by any fiber.  
Up-search for "corner Fiber f3", down-search otherwise.

write "0" into adr "oldmem" to mark FIFO available

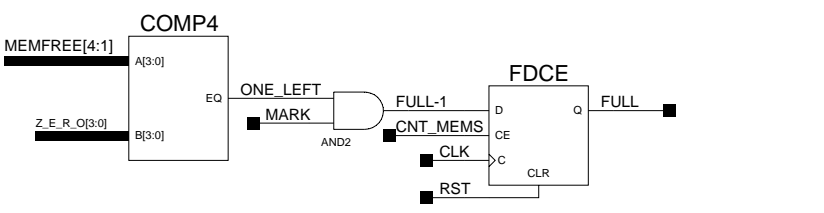
# Assign Fiber to FIFO & Assign FIFO to Fiber



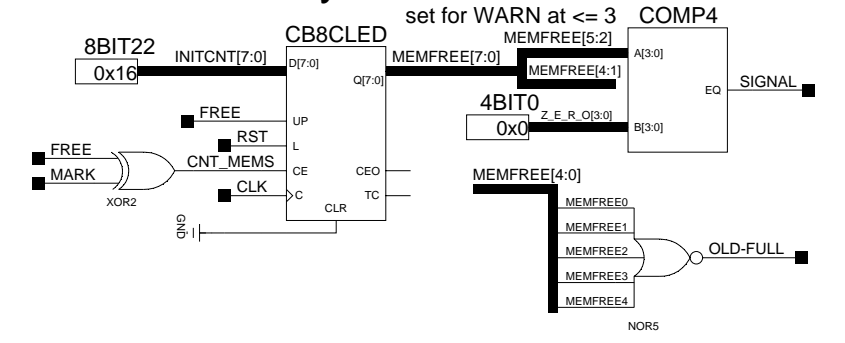
# FIFO Assign Control

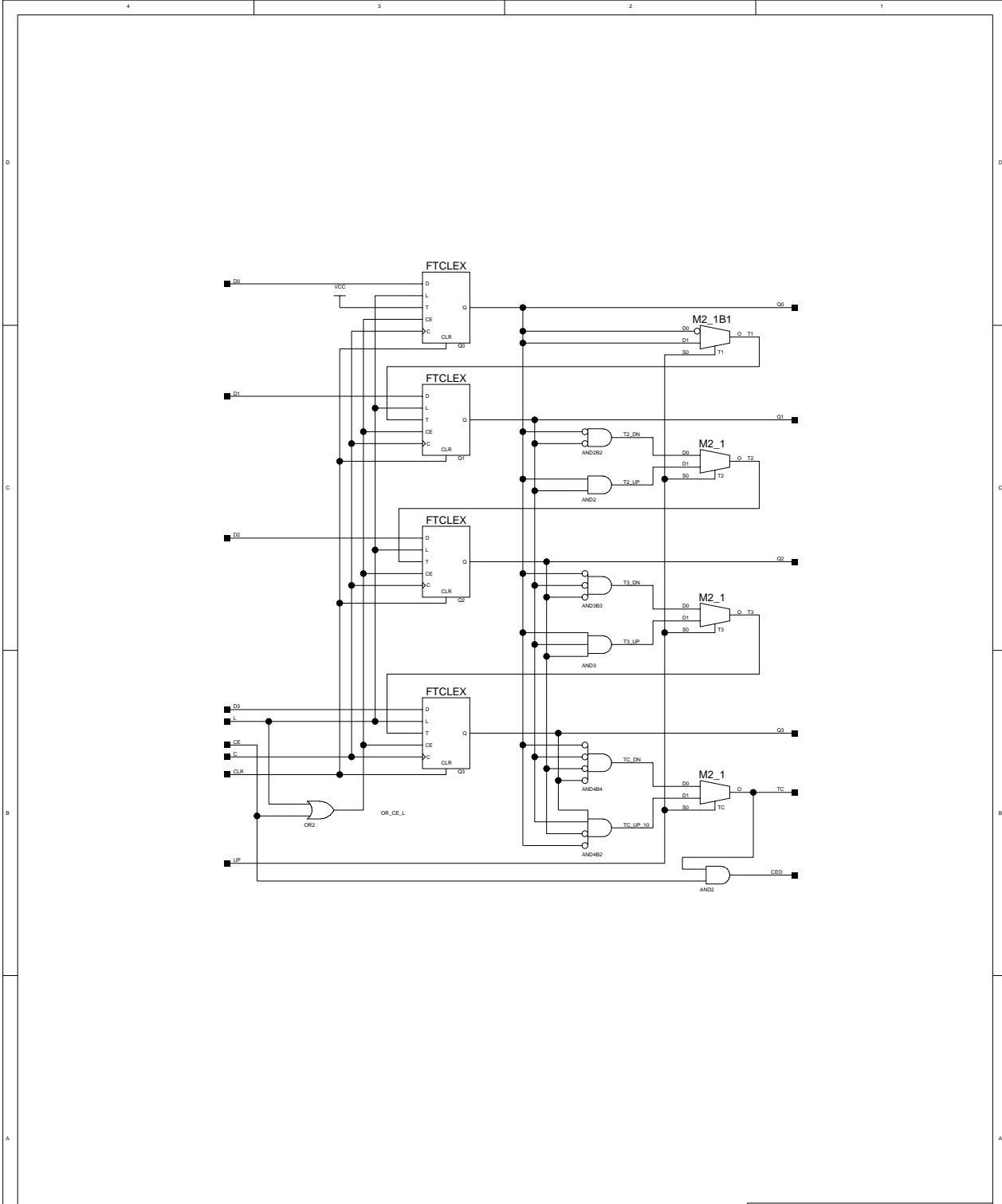


PREF\_C[1:0]  
BUSY\_C[1:0]  
GOT\_C[1:0]



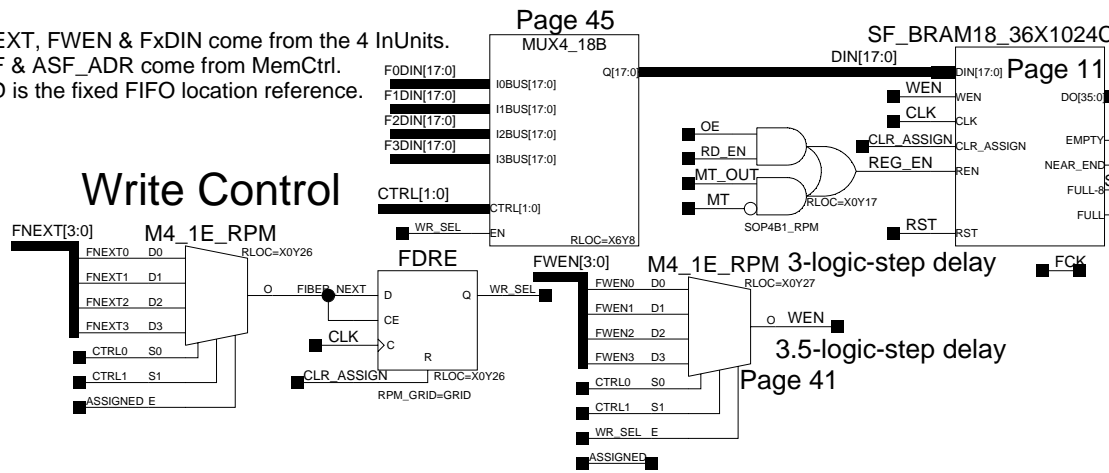
# How many of 22 FIFOs are free?



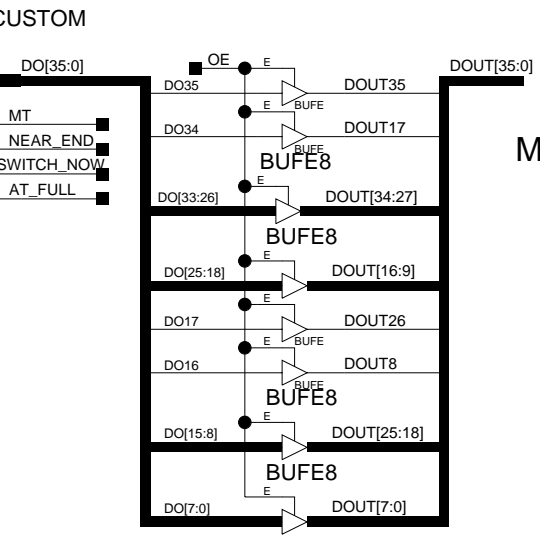


FNEXT, FWEN & FxDIN come from the 4 InUnits.  
 ASF & ASF\_ADR come from MemCtrl.  
 FAD is the fixed FIFO location reference.

### Write Control

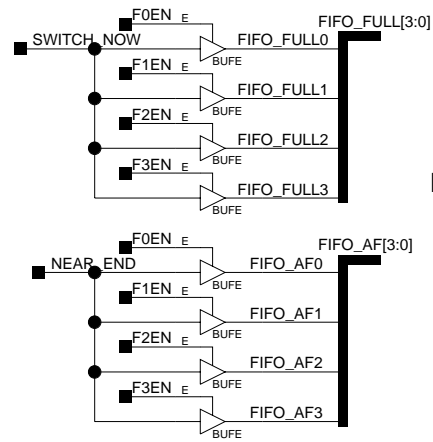
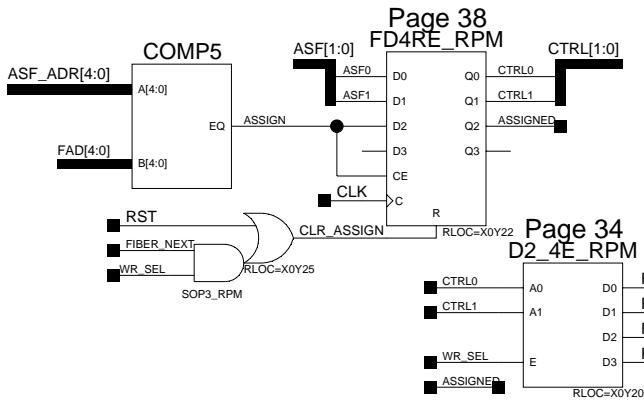


Page 11

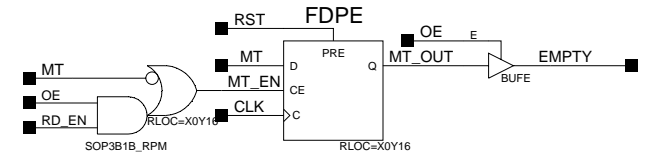


### MemIn-MemOut Mapping:

DO35	Dout35
DO34	Dout17
DO33-26	Dout34-27
DO25-18	Dout16-9
DO17	Dout26
DO16	Dout8
DO15-8	Dout25-18
Din7-0	Dout7-0



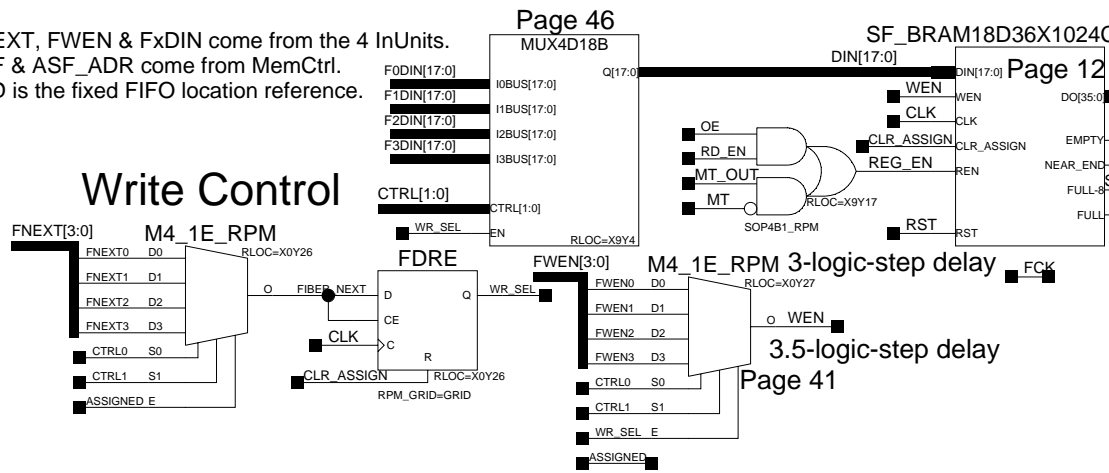
### First Word Fall Through logic:



Not Used: REN = OE \* RD\_EN

FNEXT, FWEN & FxDIN come from the 4 InUnits.  
 ASF & ASF\_ADR come from MemCtrl.  
 FAD is the fixed FIFO location reference.

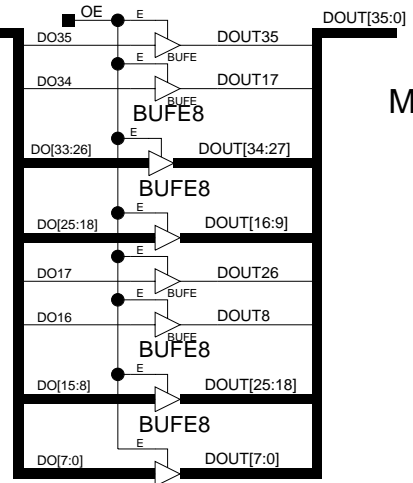
### Write Control



Page 46

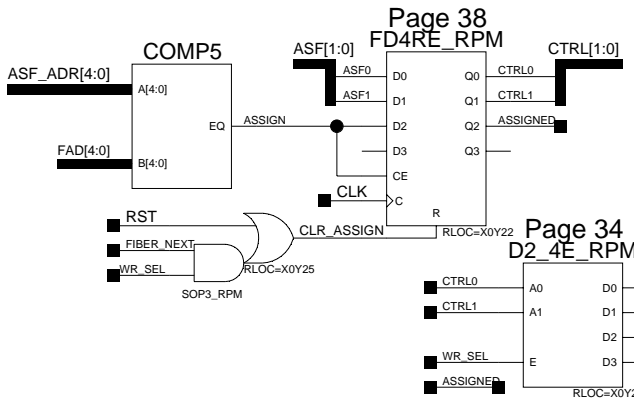
SF\_BRAM18D36X1024CUSTOM

Page 12



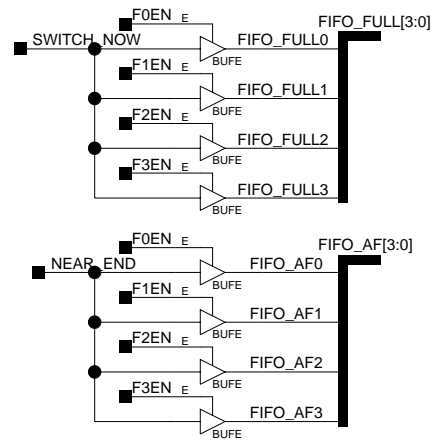
### MemIn-MemOut Mapping:

DO35	Dout35
DO34	Dout17
DO33-26	Dout34-27
DO25-18	Dout16-9
DO17	Dout26
DO16	Dout8
DO15-8	Dout25-18
Din7-0	Dout7-0

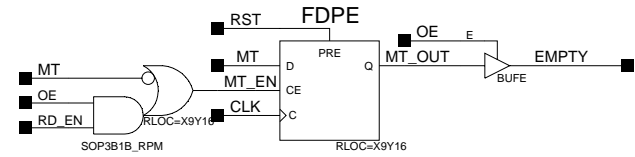


Page 38

Page 34



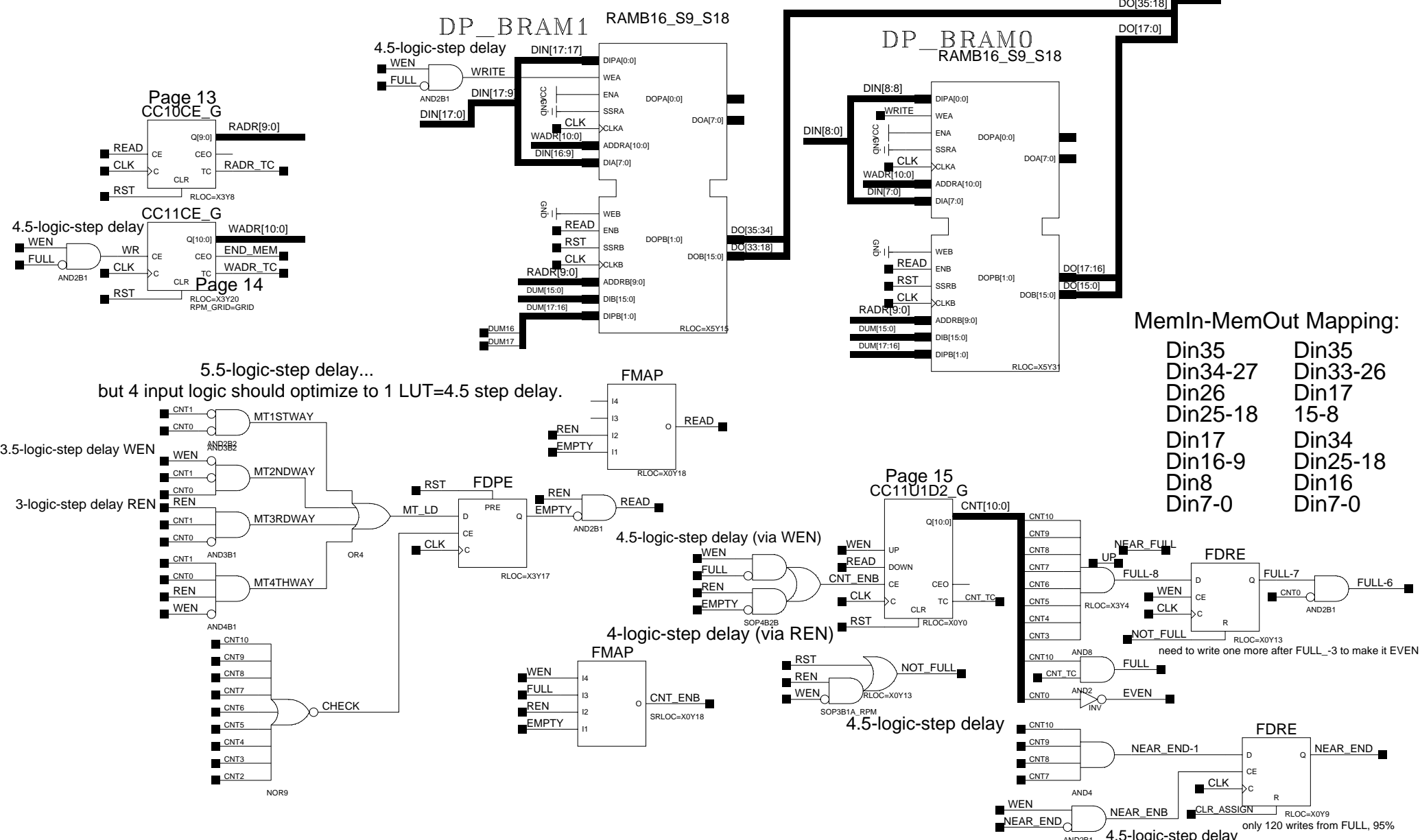
### First Word Fall Through logic:



Not Used: REN = OE \* RD\_EN



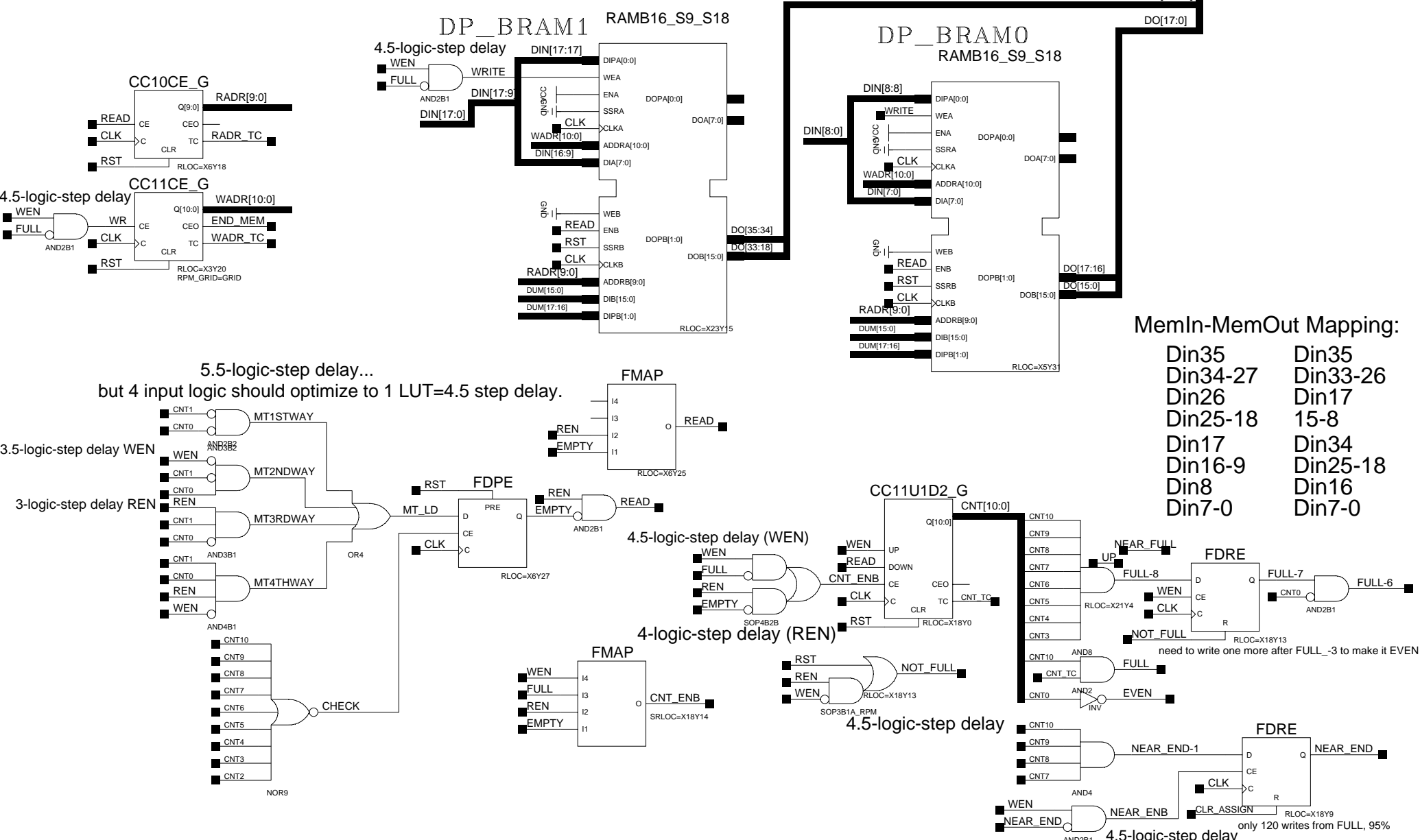
# sfifo18\_36x1024



**MemIn-MemOut Mapping:**

Din35	Din35
Din34-27	Din33-26
Din26	Din17
Din25-18	15-8
Din17	Din34
Din16-9	Din25-18
Din8	Din16
Din7-0	Din7-0

# sfifo18\_36x1024\_diagonal



**MemIn-MemOut Mapping:**

Din35	Din35
Din34-27	Din33-26
Din26	Din17
Din25-18	15-8
Din17	Din34
Din16-9	Din25-18
Din8	Din16
Din7-0	Din7-0

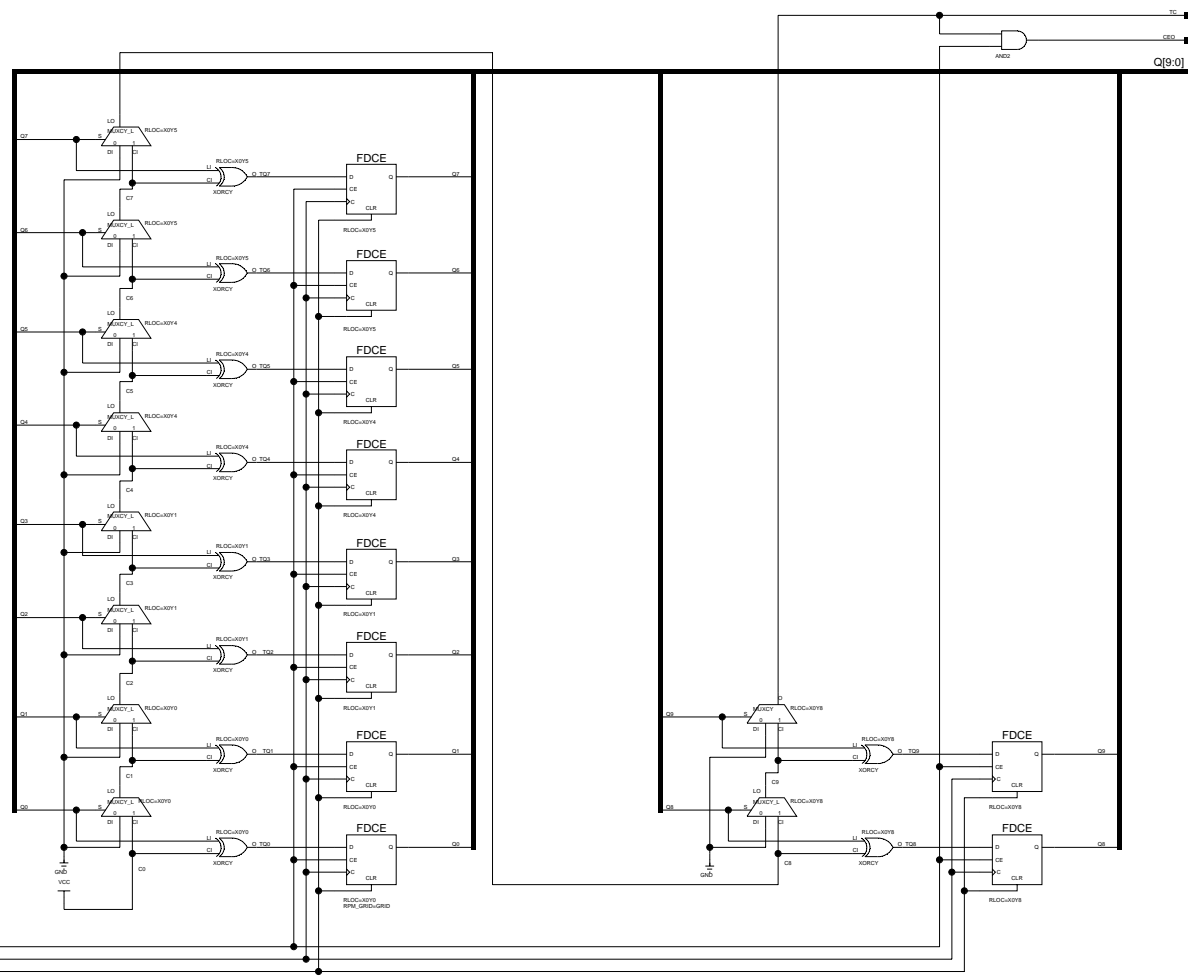
5.5-logic-step delay...  
but 4 input logic should optimize to 1 LUT=4.5 step delay.

4.5-logic-step delay (WEN)

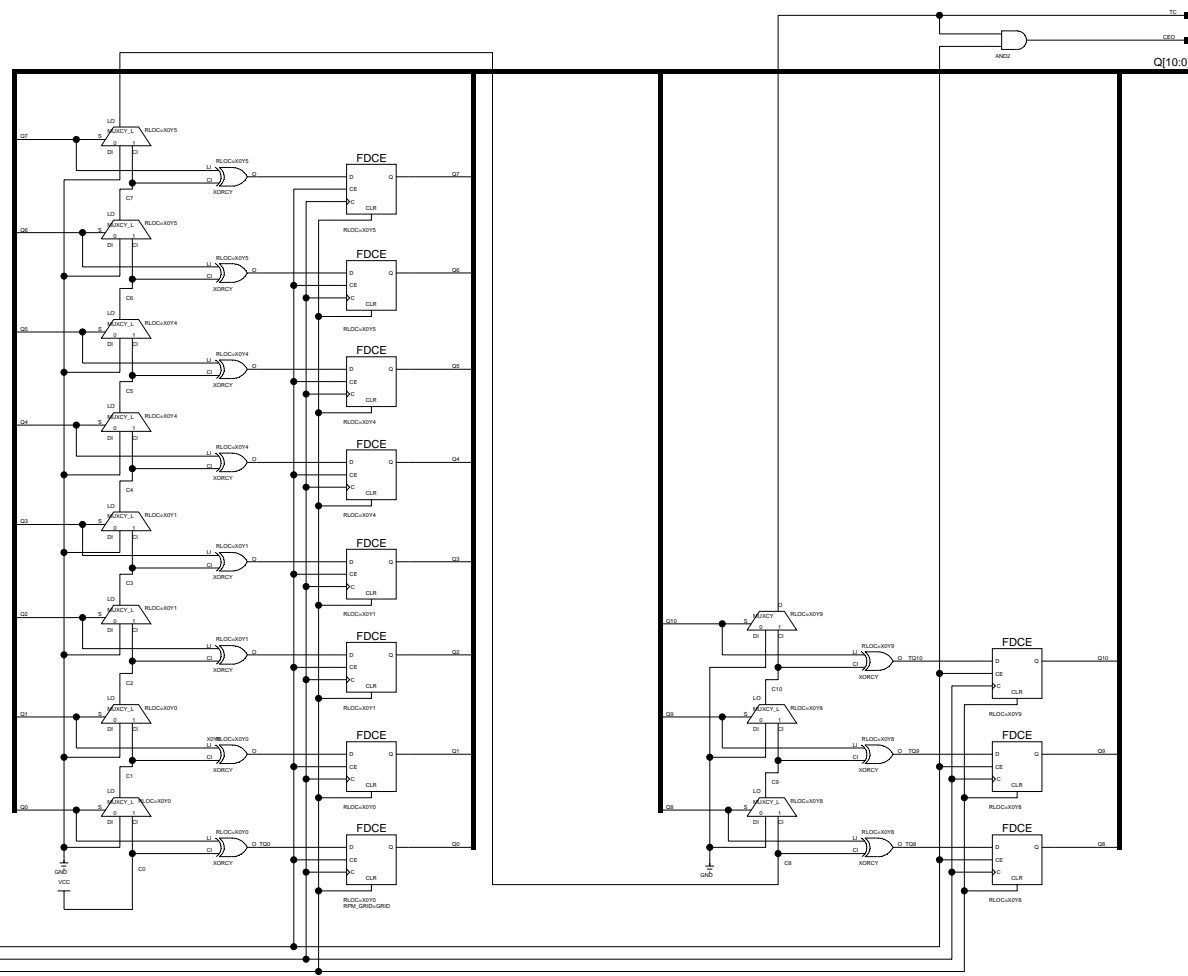
4-logic-step delay (REN)

4.5-logic-step delay

4.5-logic-step delay

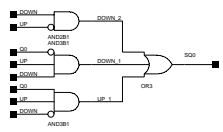
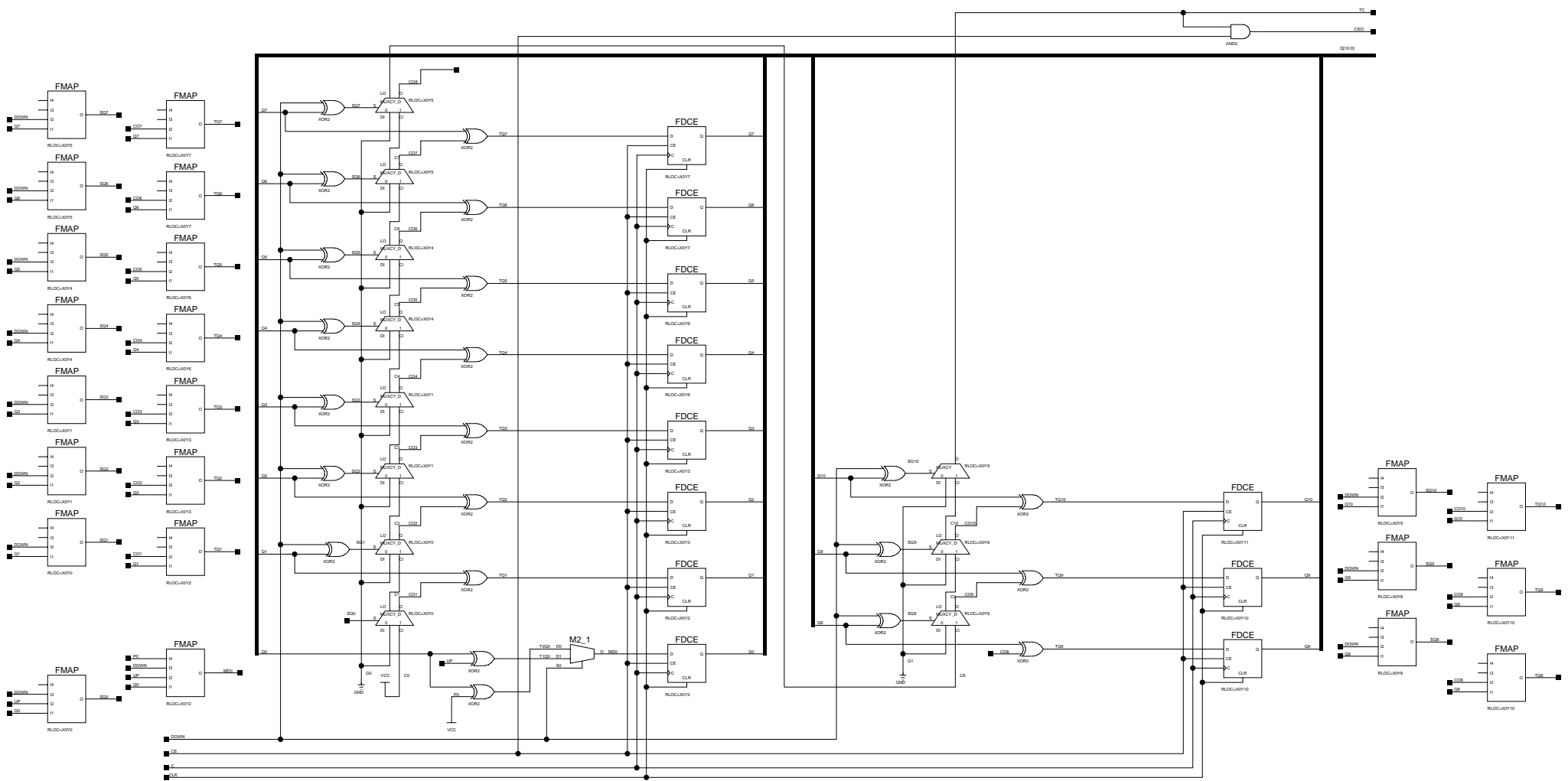


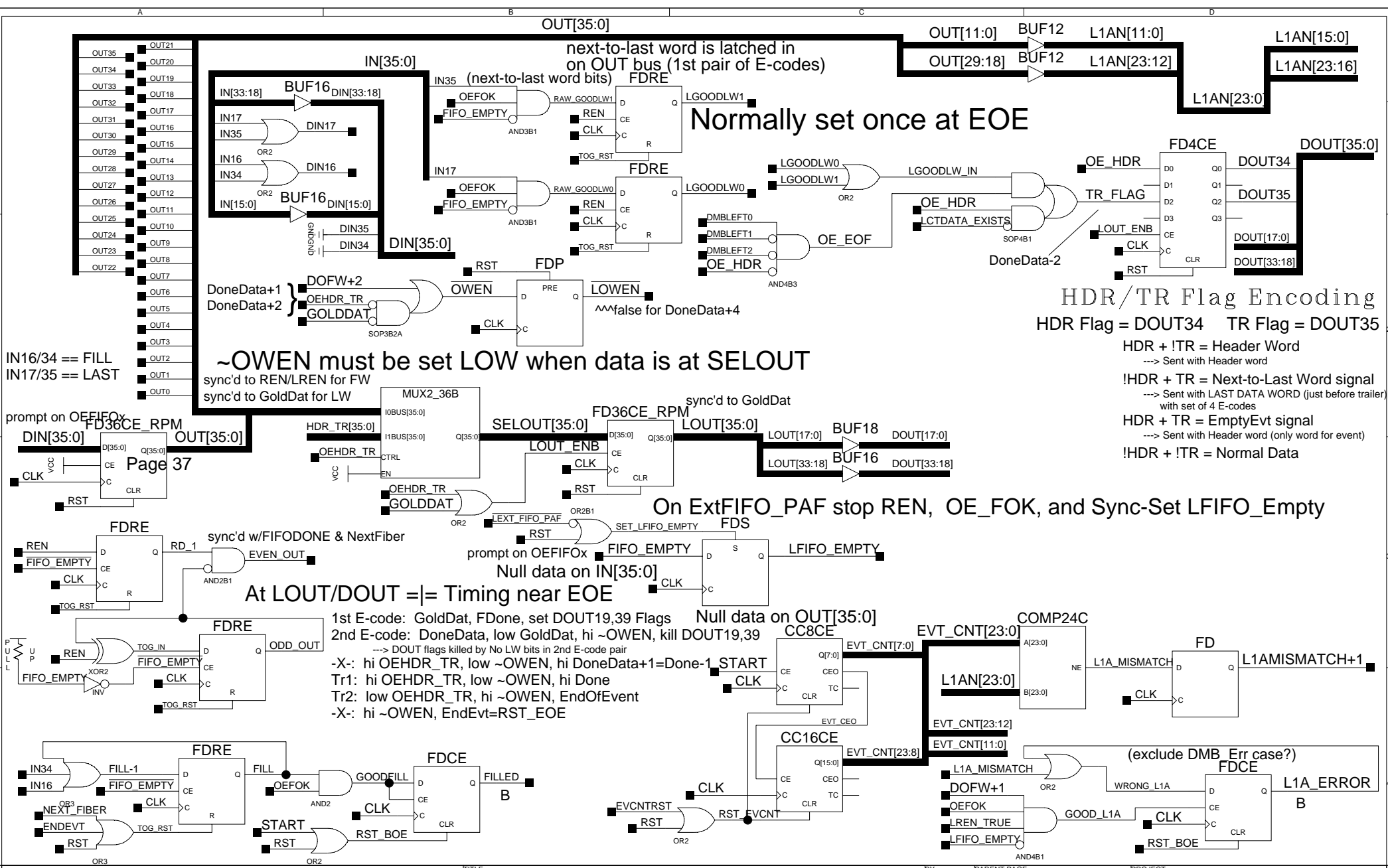
■ CE  
 ■ C  
 ■ Q



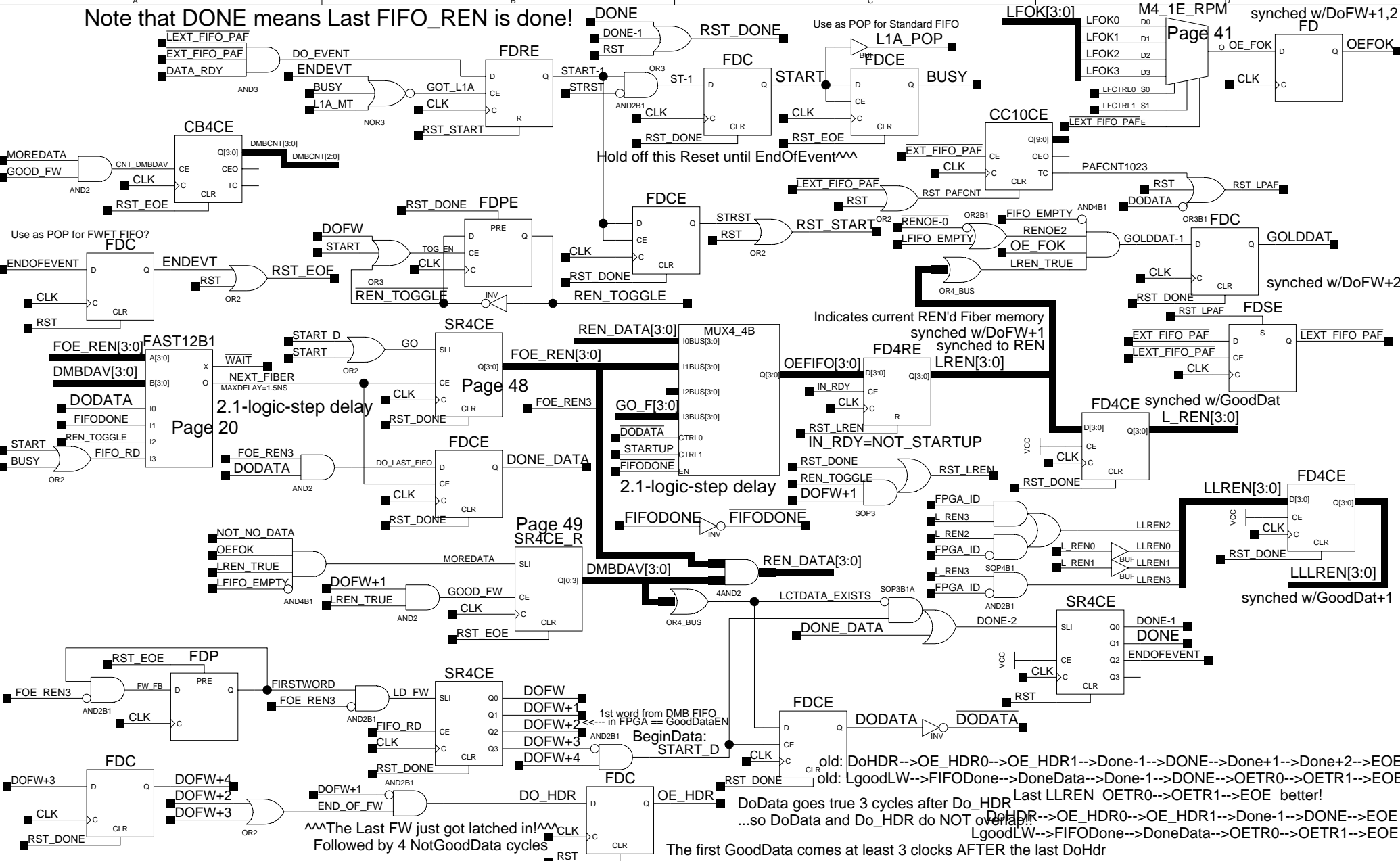
■ CE  
 ■ C  
 ■ CLR

UP = Increment 1  
 UP.DOWN = Decrement 1  
 DOWN = Decrement 2  
 Assume that CE includes DOWN+UP

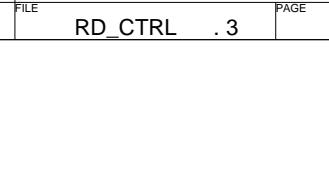
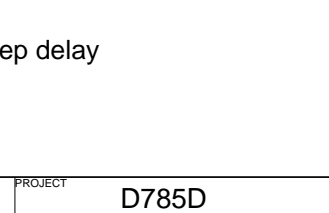
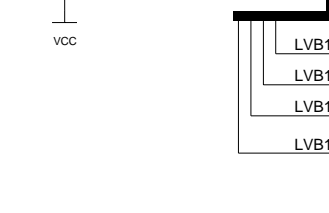
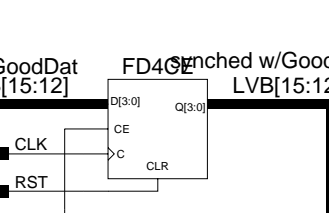
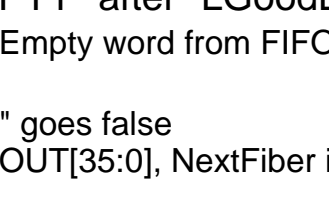
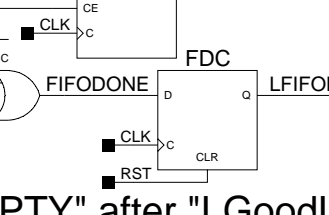
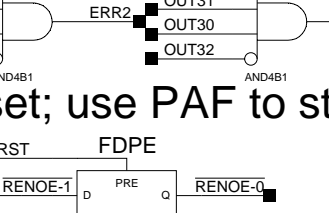
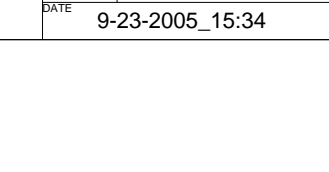
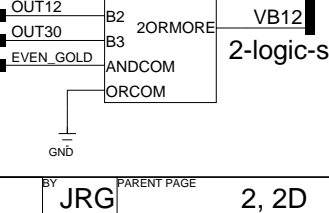
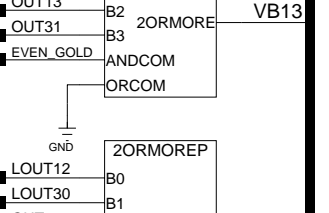
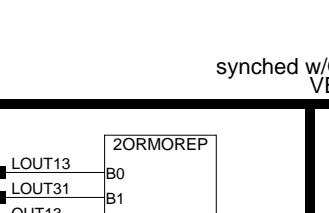
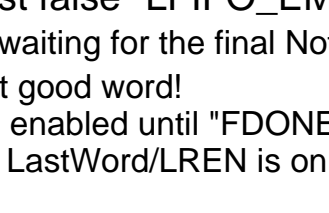
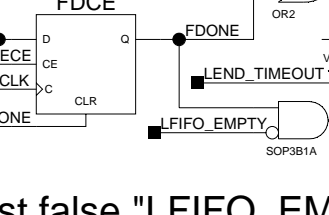
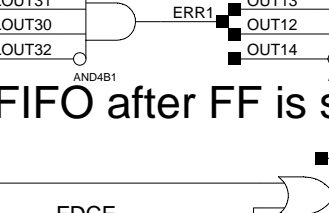
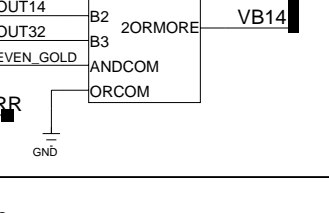
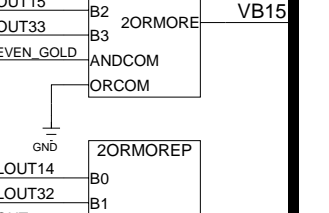
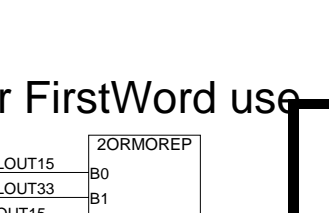
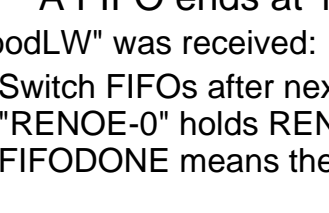
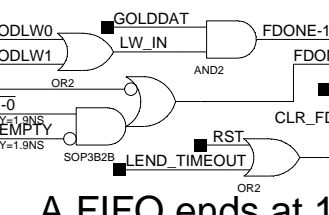
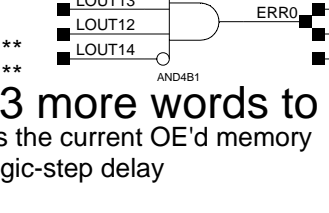
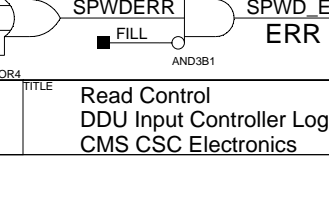
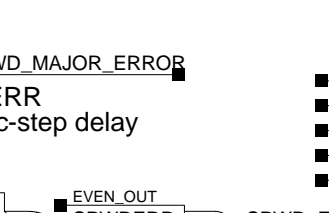
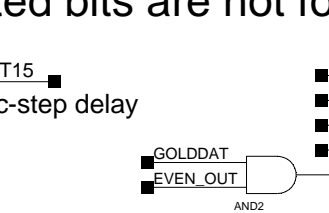
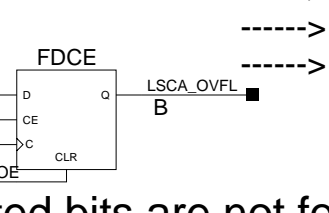
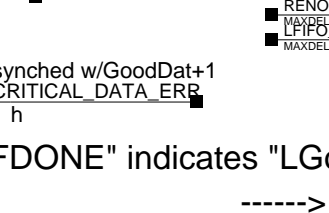
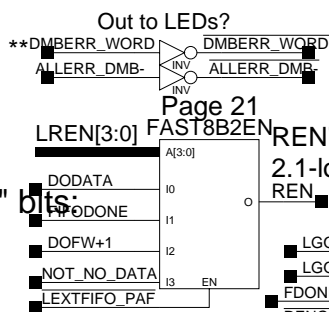
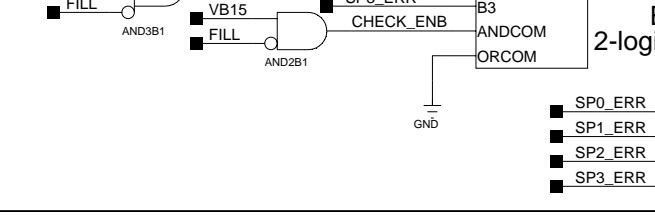
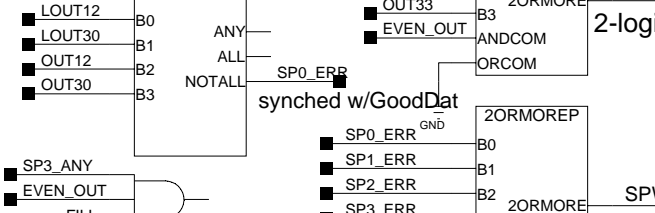
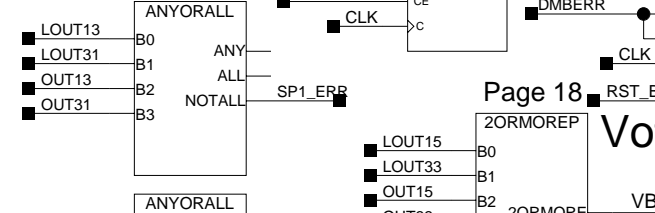
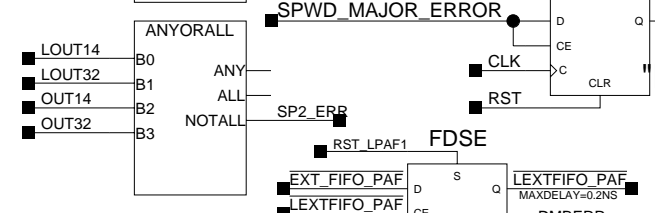
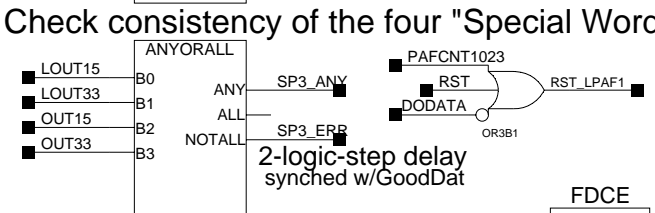
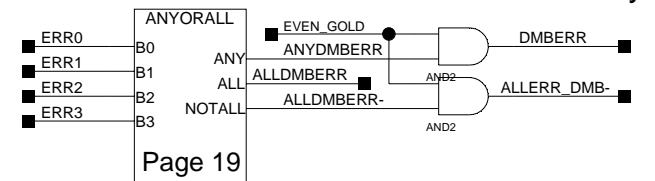




# 3 more words to FIFO after FF is set; use LEXT\_FIFO\_PAF to stop REN/OWEN!



**Check for DMB Error Word and consistency:**



**3 more words to FIFO after FF is set; use PAF to stop!**

REN's the current OE'd memory  
2.1-logic-step delay

**A FIFO ends at 1st false "LFIFO\_EMPTY" after "LGoodLW"**

"FDONE" indicates "LGoodLW" was received: waiting for the final NotEmpty word from FIFO

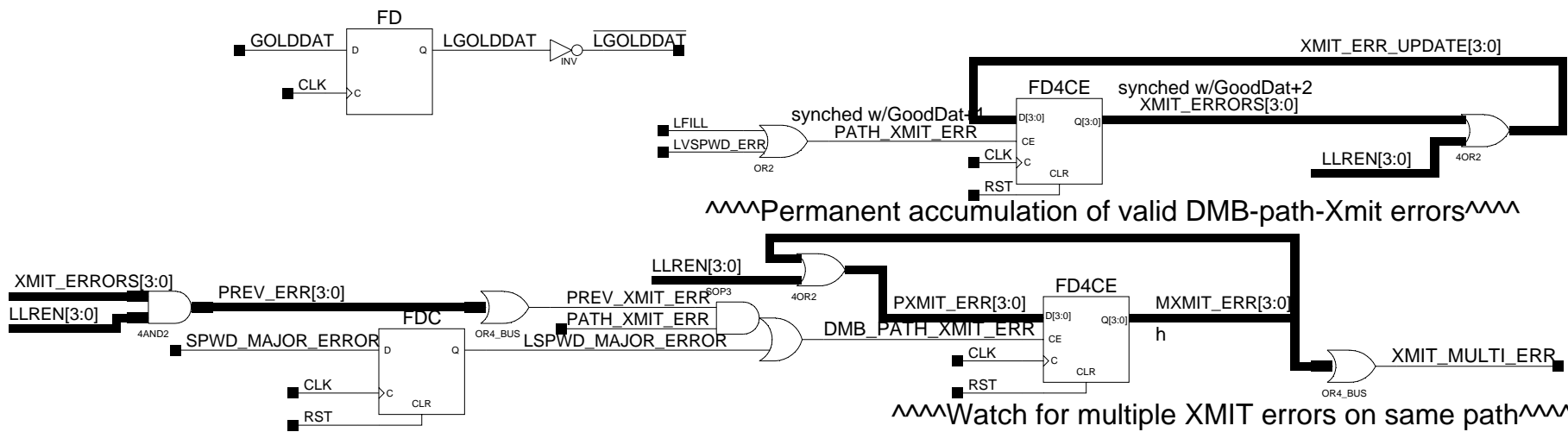
-----> Switch FIFOs after next good word!

-----> "RENOE-0" holds REN enabled until "FDONE" goes false

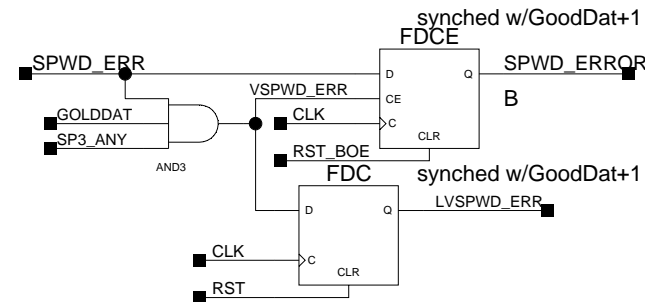
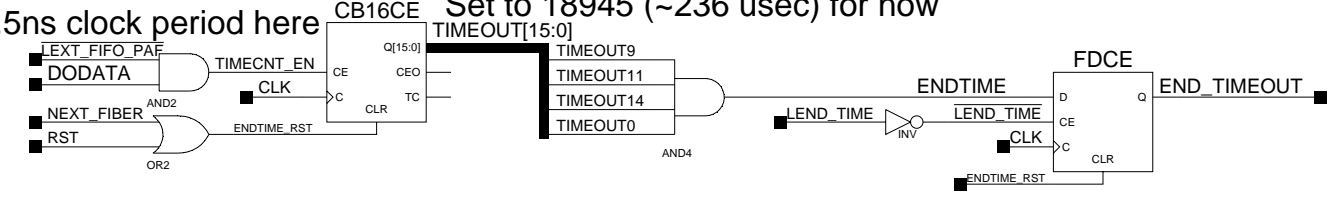
-----> FIFODONE means the LastWord/LREN is on OUT[35:0], NextFiber is set

**Voted bits are not for FirstWord use**



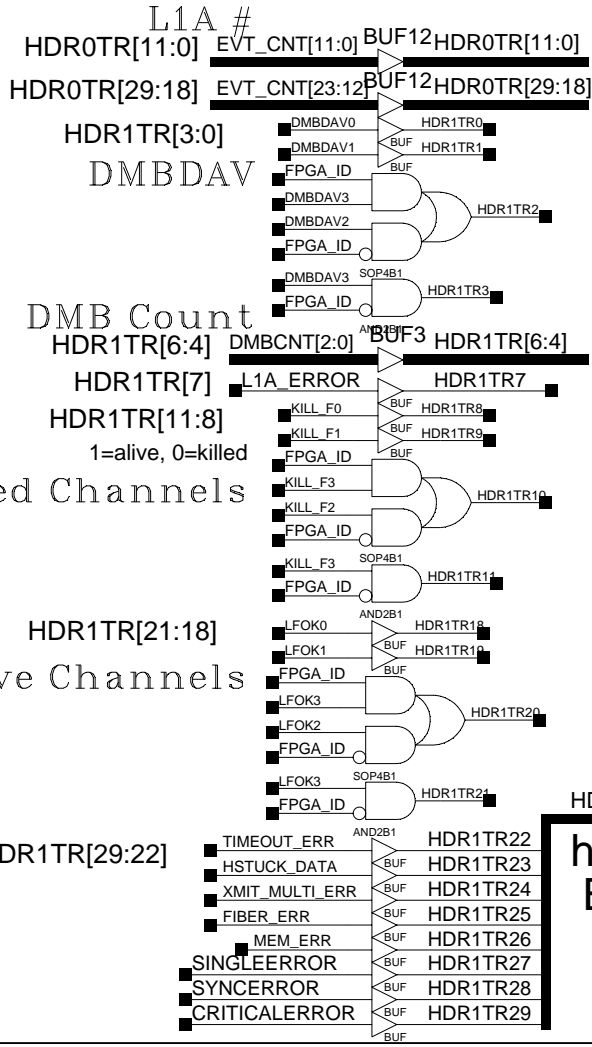


FIFO Done Timeout: count to 10752, 132 usec=10625 is the worst case possible per FIFO?  
 Set to 18945 (~236 usec) for now

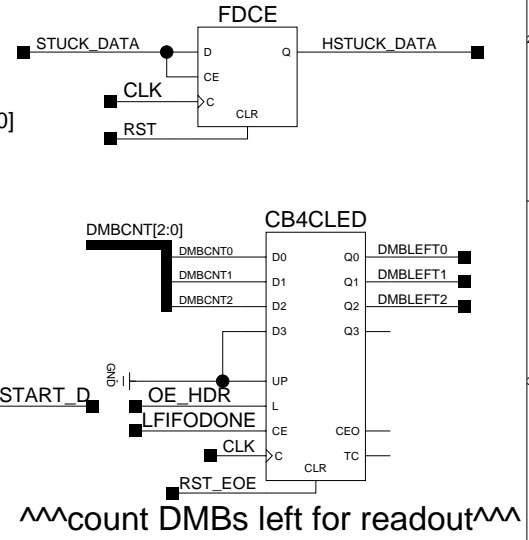
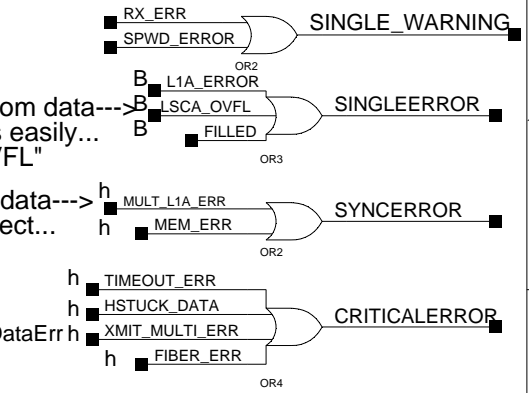
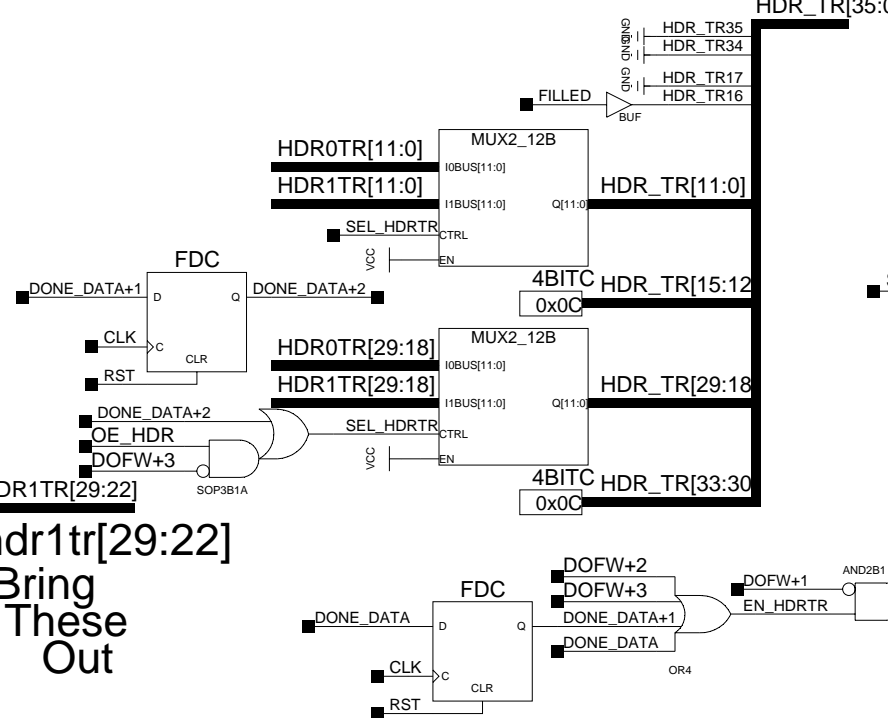


# Status Monitor

Pick 8 Status signals, include in DAQ HDR/TR  
 -create logic for OEHDR/TR and their format  
 Also set 4 FMM bits for each input channel



Note: all HDR\_TR[35:34,17:16] bits are free  
 also for HDR0TR & HDR1TR  
 - use for Event Type/Pasthrough flag?  
 - use for additional error reporting?



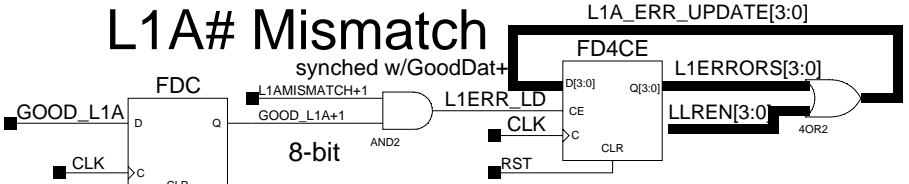
Killed Channels

Live Channels

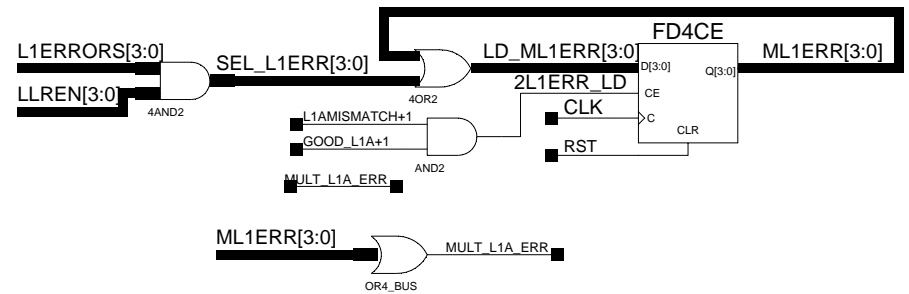
Bring These Out

^^count DMBs left for readout^^

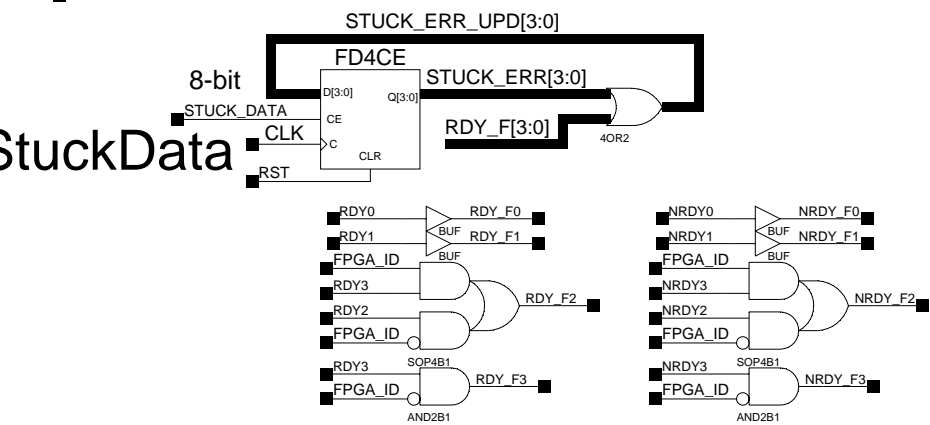
# L1A# Mismatch



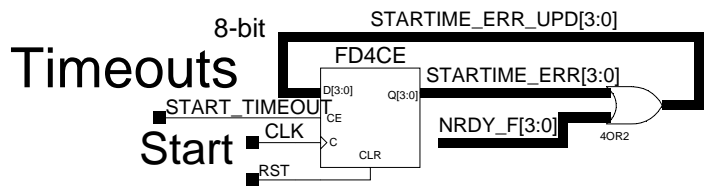
# Multi-L1A# Mismatch



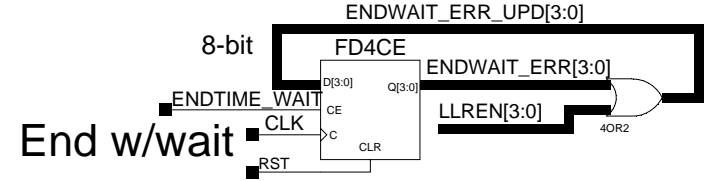
# StuckData



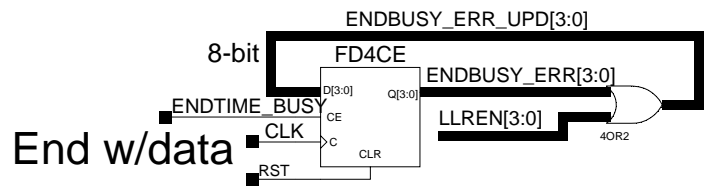
# Timeouts



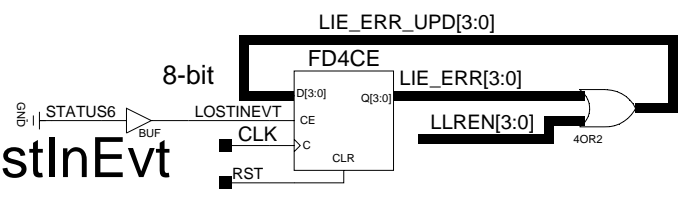
# End w/wait



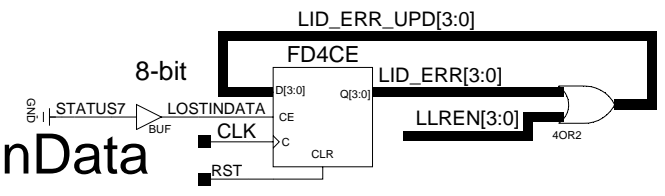
# End w/data



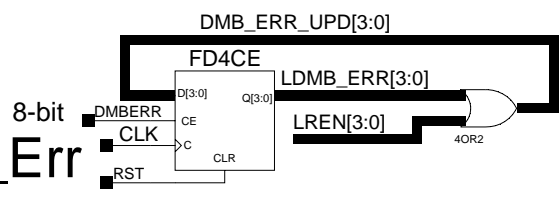
# LostInEvt



# LostInData



# DMB\_Err



H1: 0x/5T/NN.NNNN/XXX/I.II/VK

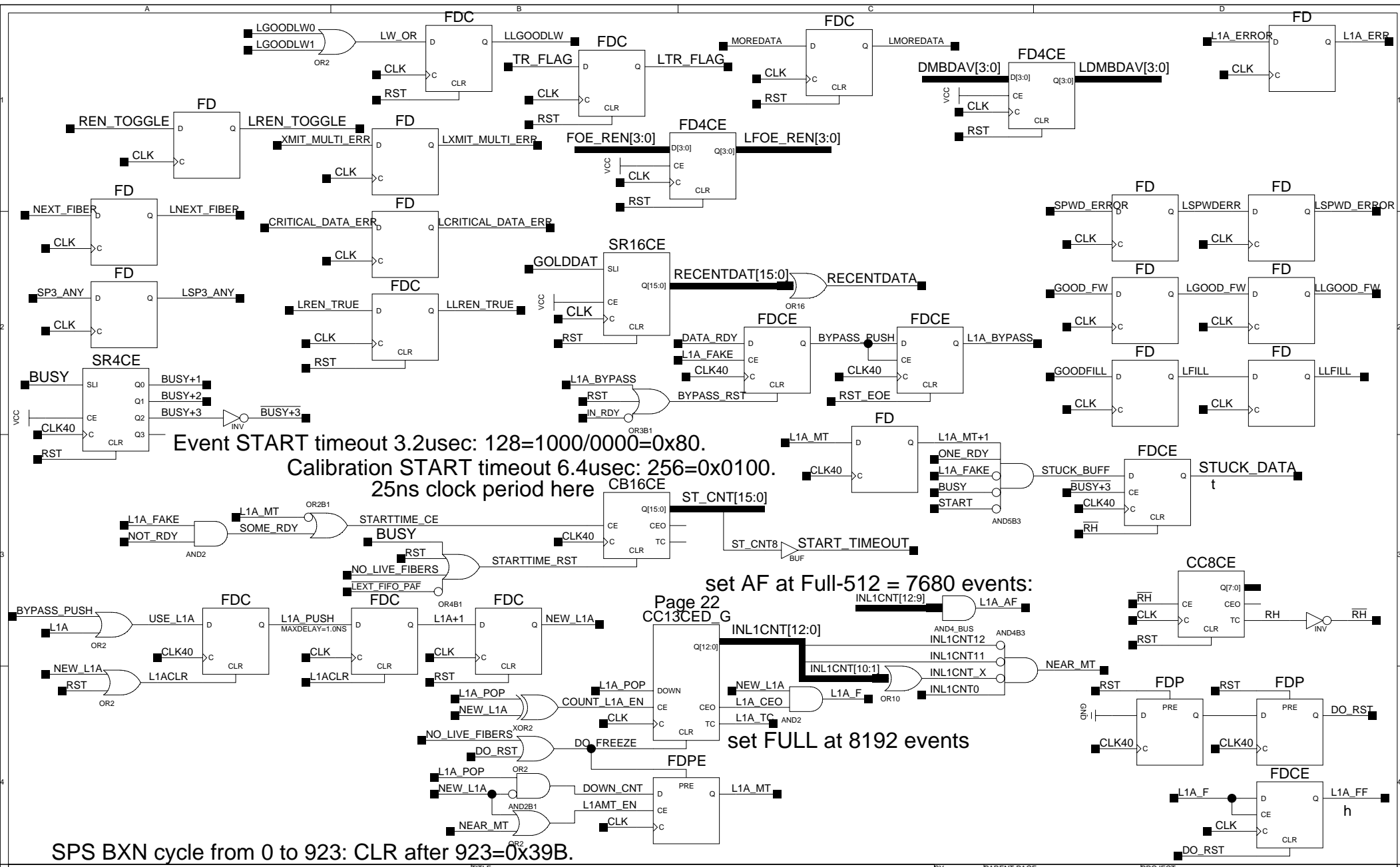
H2: 0x/8000/0001/8000/HHHH

H3: 0x/LLLL/oooo/ZZZZ/GGMY

T-2: 0x/8000/FFFF/8000/8000

T-1: 0x/SSSS.SSSS/QQQQ/PPPP

TR: 0x/A/?/WW.WWWW/RRRR/UUMK



Event START timeout 3.2usec: 128=1000/0000=0x80.  
 Calibration START timeout 6.4usec: 256=0x0100.  
 25ns clock period here

set AF at Full-512 = 7680 events:

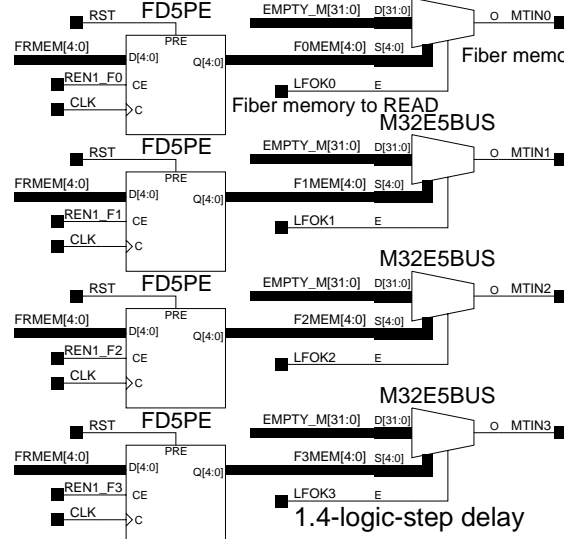
set FULL at 8192 events

SPS BXN cycle from 0 to 923: CLR after 923=0x39B.

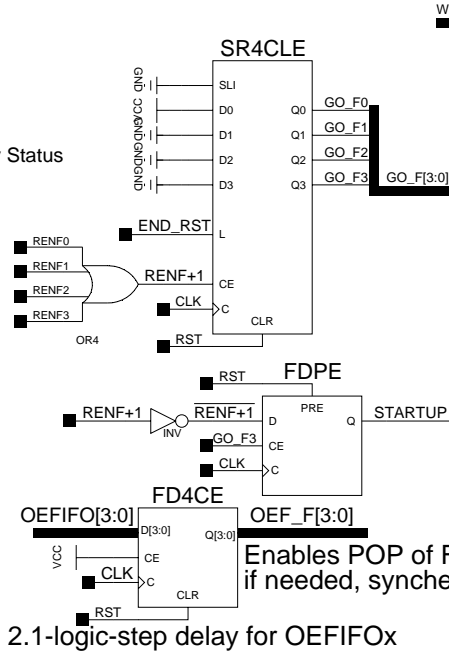
Page 22  
 CC13CED G

END

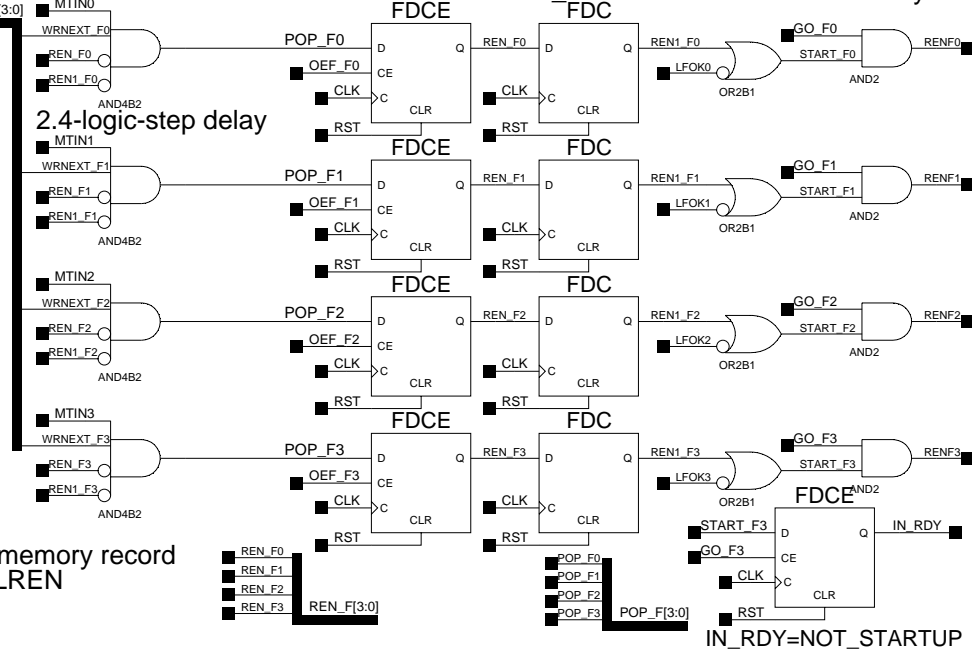
Page 25  
M32E5BUS



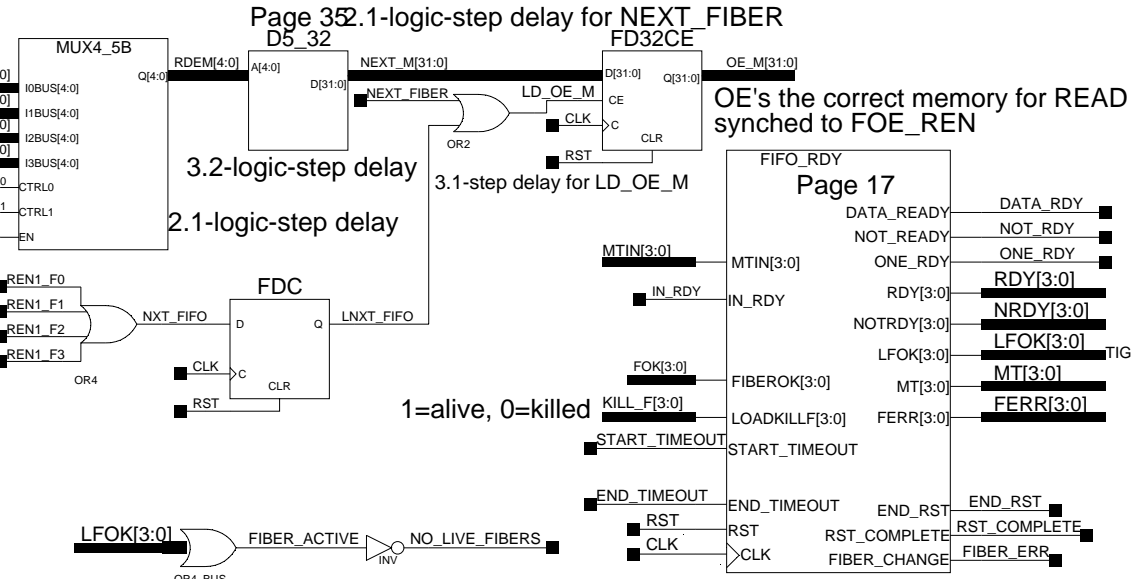
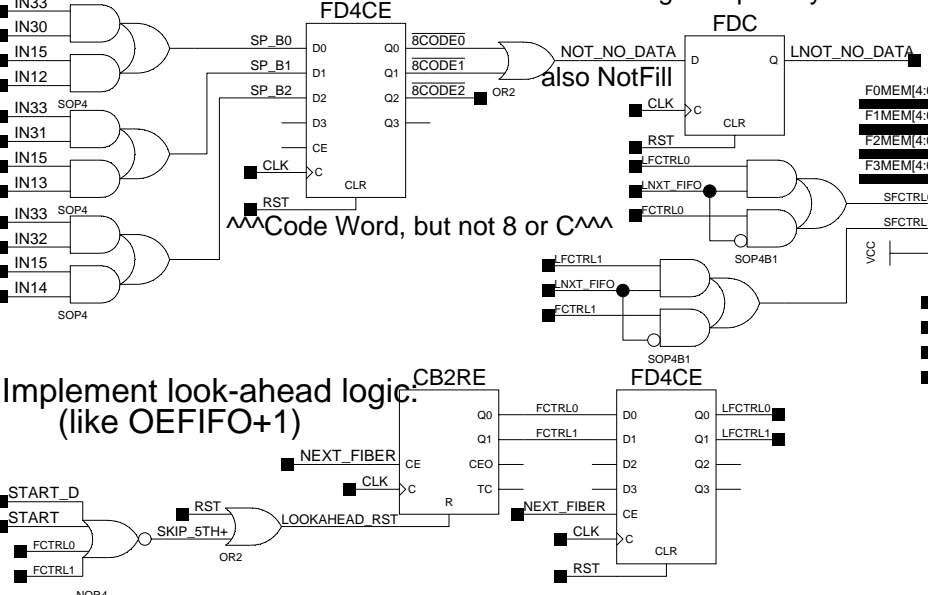
WRNEXT\_Fx: Fiber is writing to next memory

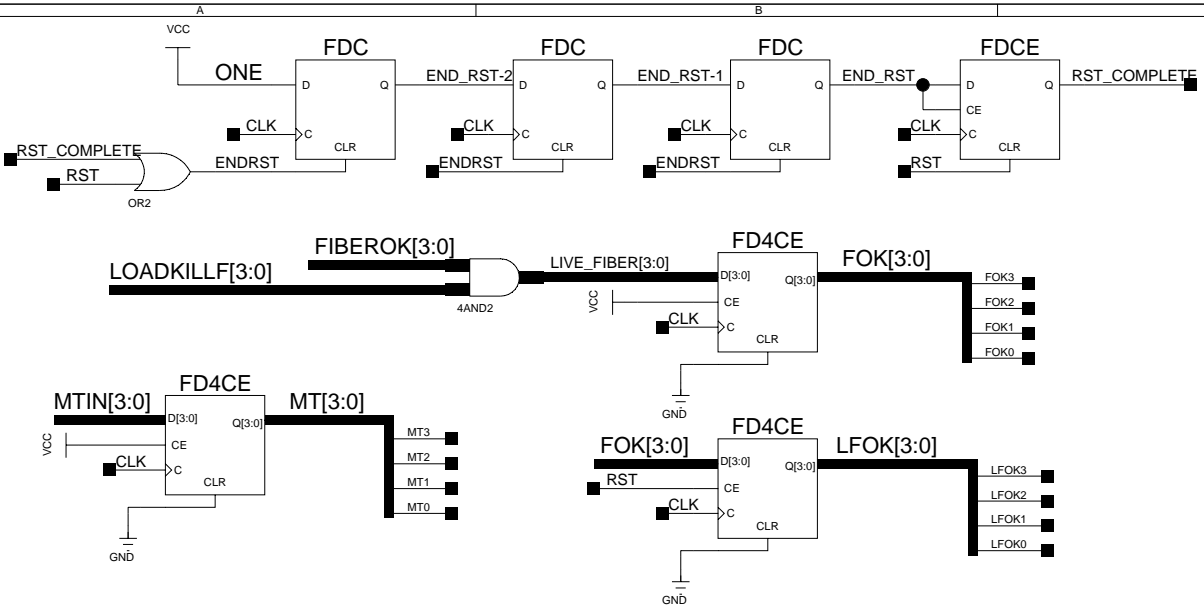


REN\_Fx used to POP Fiber memory record

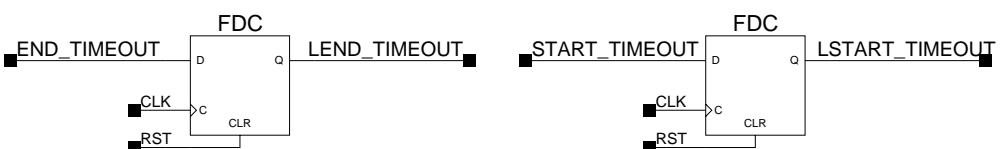


Page 32.1-logic-step delay for NEXT\_FIBER

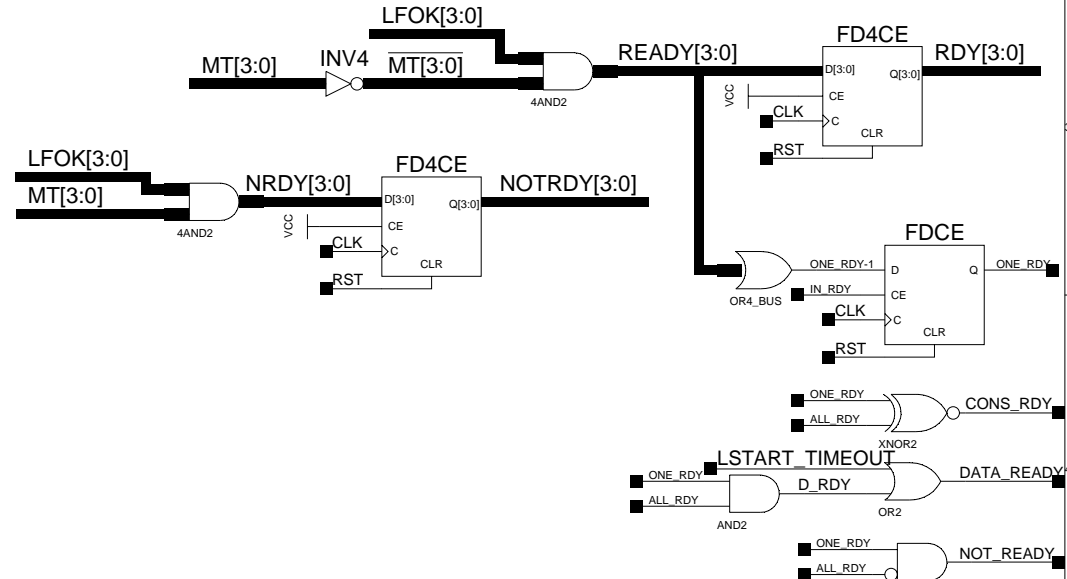


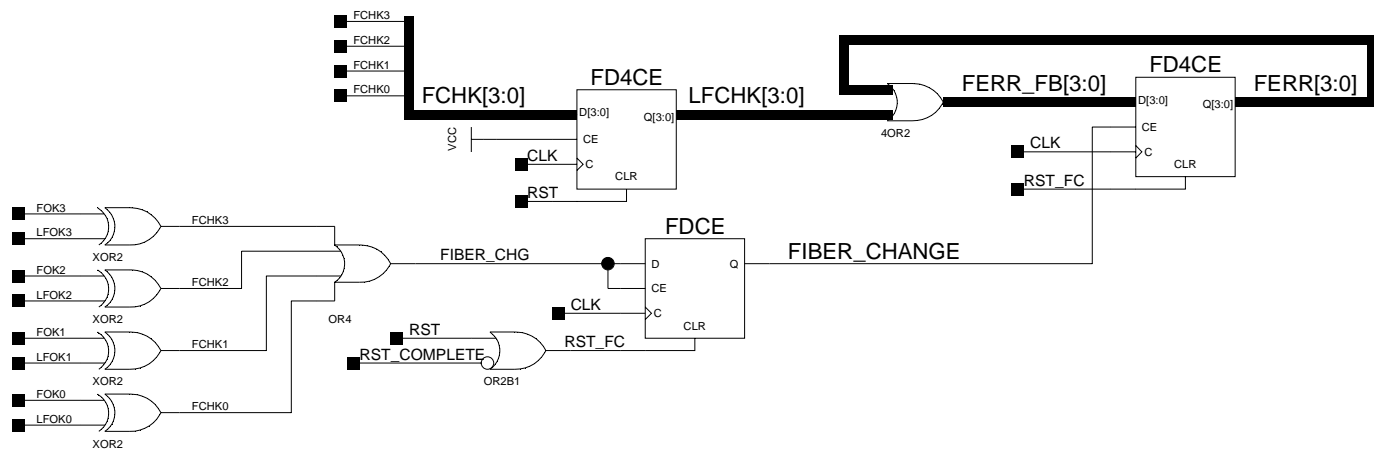


Access FIBEROK from JTAG as a fiber check.  
 ---> RESET required after fiber status change for now...  
 Change in FIBEROK is BAD! Set error code.  
 ...notify FMM and maybe set STATUS bit, but  
 ...data is OK until FIFO read time-out occurs.  
 ...but how to know WHEN the bad-data comes out  
 ---> timeout will probably occur for that event

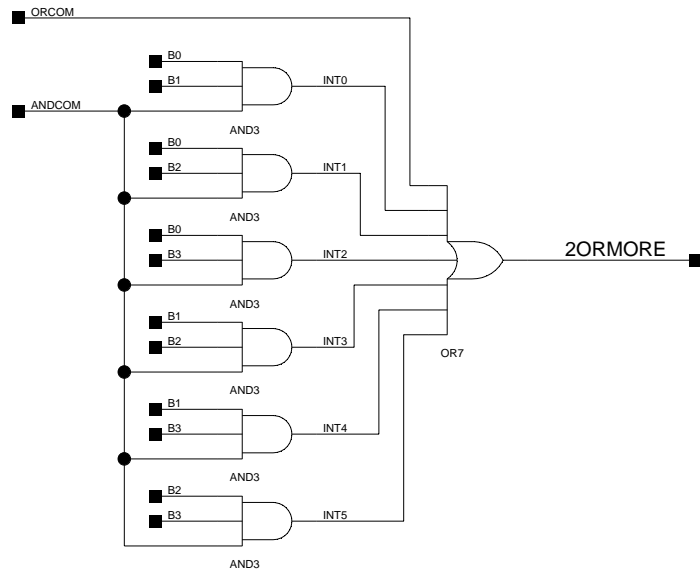


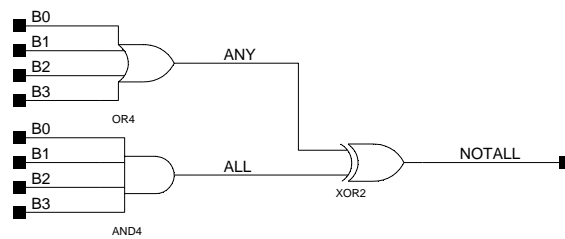
MASK fiber on Start/End TIMEOUT? Kill it in LFOK...

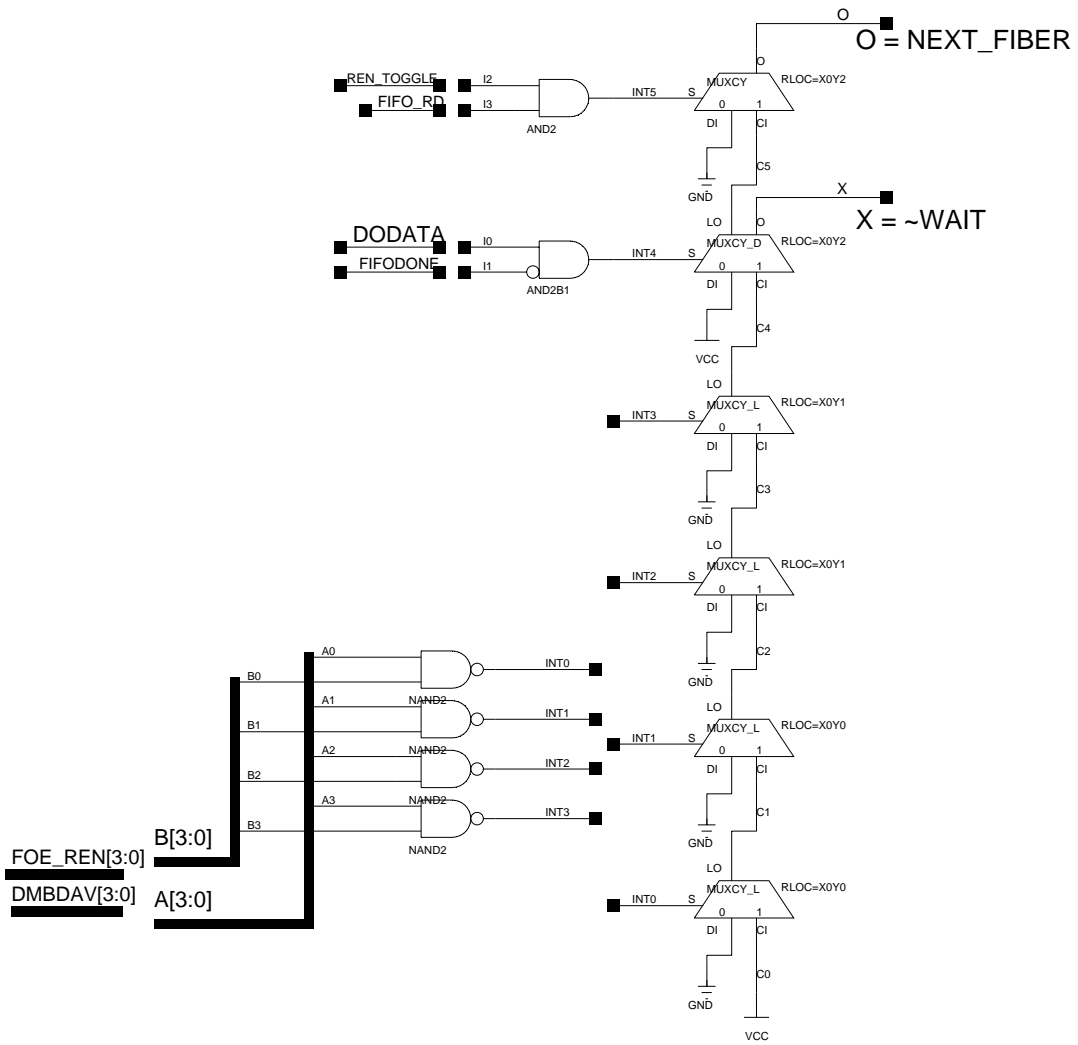




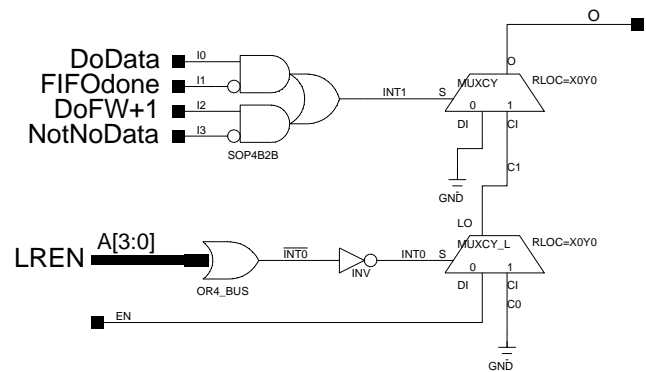








Title:	FAST12B1	
Comments:	Custom Fast, Complex Logic for DDU, use 4 MUXCY as OR, 2 as AND similar to: OR of 4 AND2 .AND. AND2B1 .AND. AND2	
Date:	15th October 2003	Ver: 1
Sheet Size:	B	Rev: A

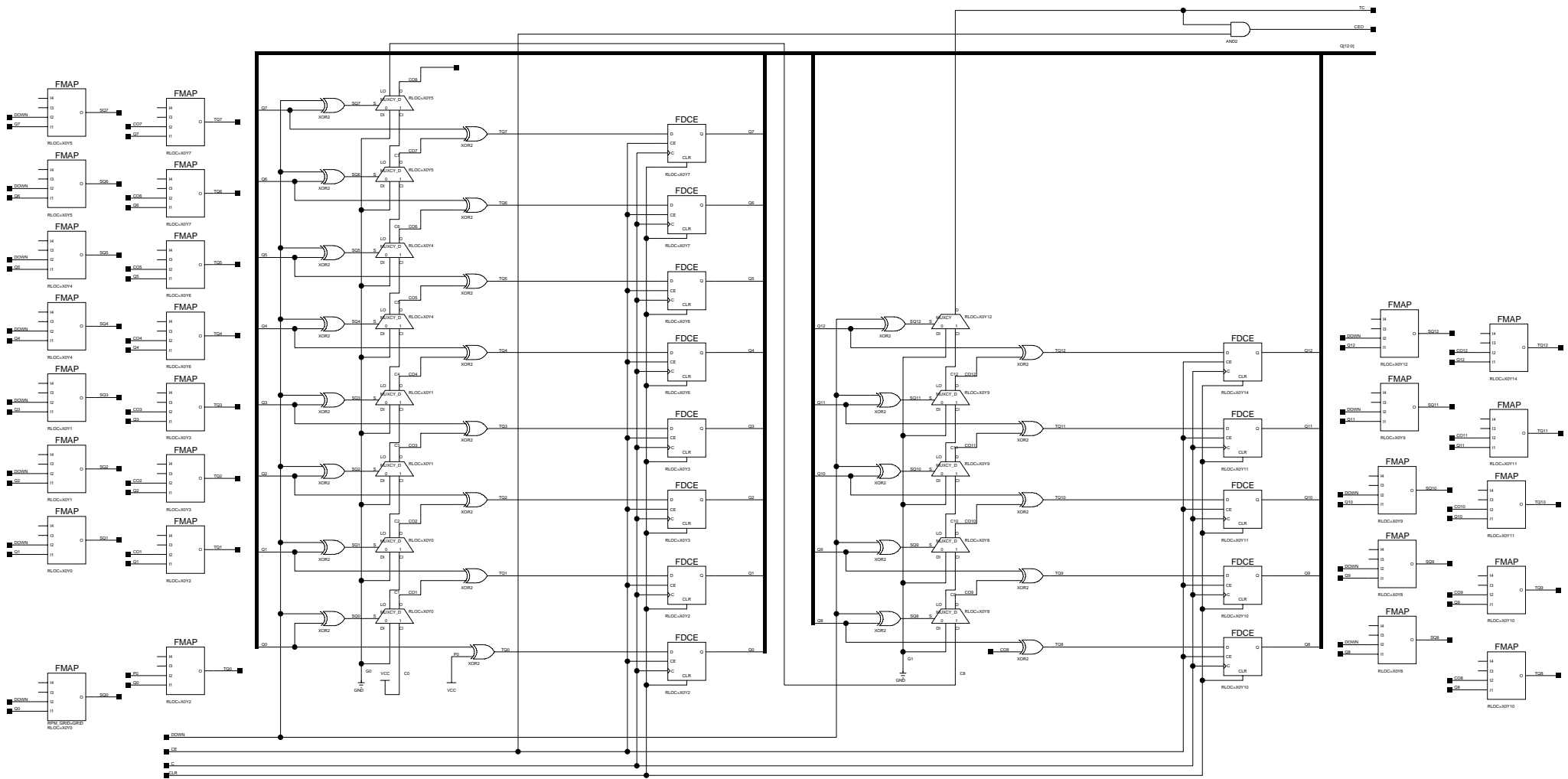


JRG

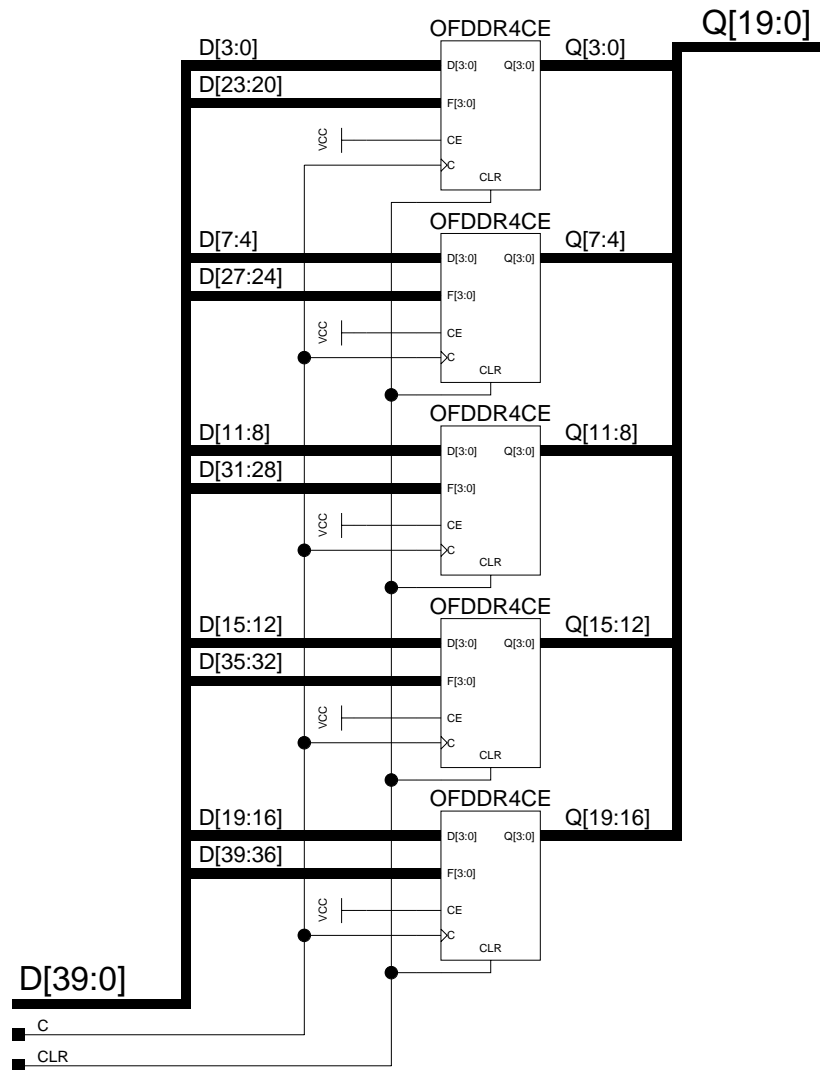
Title:	FAST8B2EN		
Comments:	Custom Fast, Complex Logic for DDU, use 2 MUXCY as AND similar to: OR4 AND SOP4B2b		
Date:	14th September 2004	Ver:	1
Sheet Size:	B	Rev:	A

DOWN = Decrement 1

Assume that CE includes DOWN+UP



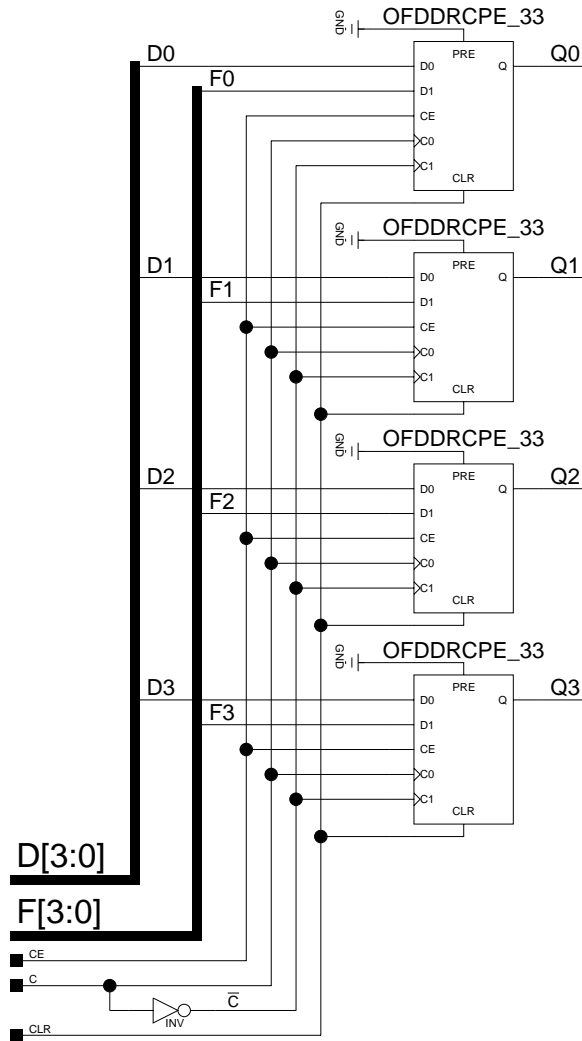
the D byte goes out last (on C rising edge)  
 the F byte goes out first (on C falling edge)

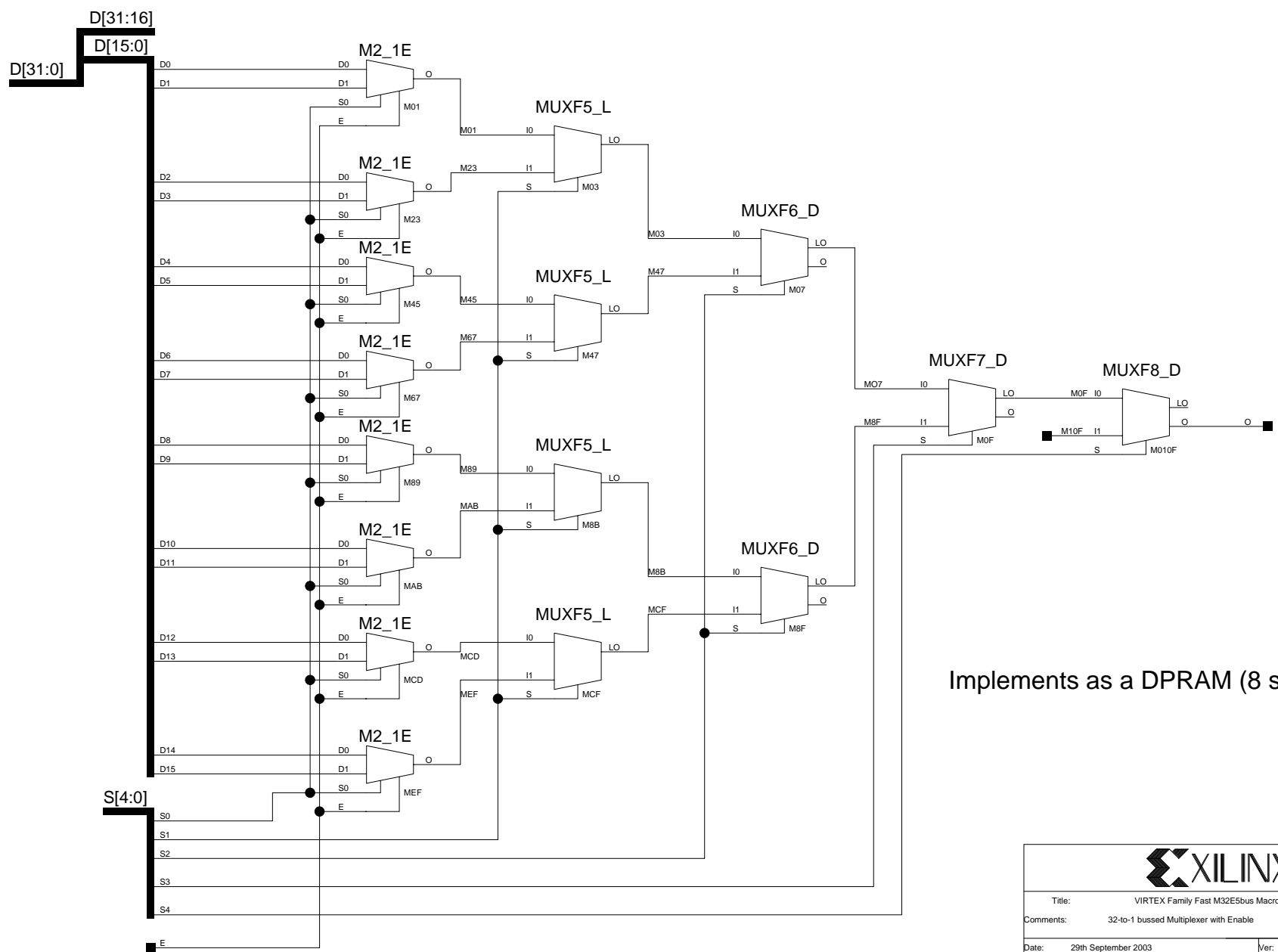


Title: VIREX Family OFDDR40 Macro	
Comment: 19-Bit Output DDR Flip-Flop w/Asynchronous Clear	
Date: 16th December 2003	Ver: 1
Sheet Size: B	Rev: A

the D byte goes out last (on C rising edge)  
 the F byte goes out first (on C falling edge)

Q[3:0]





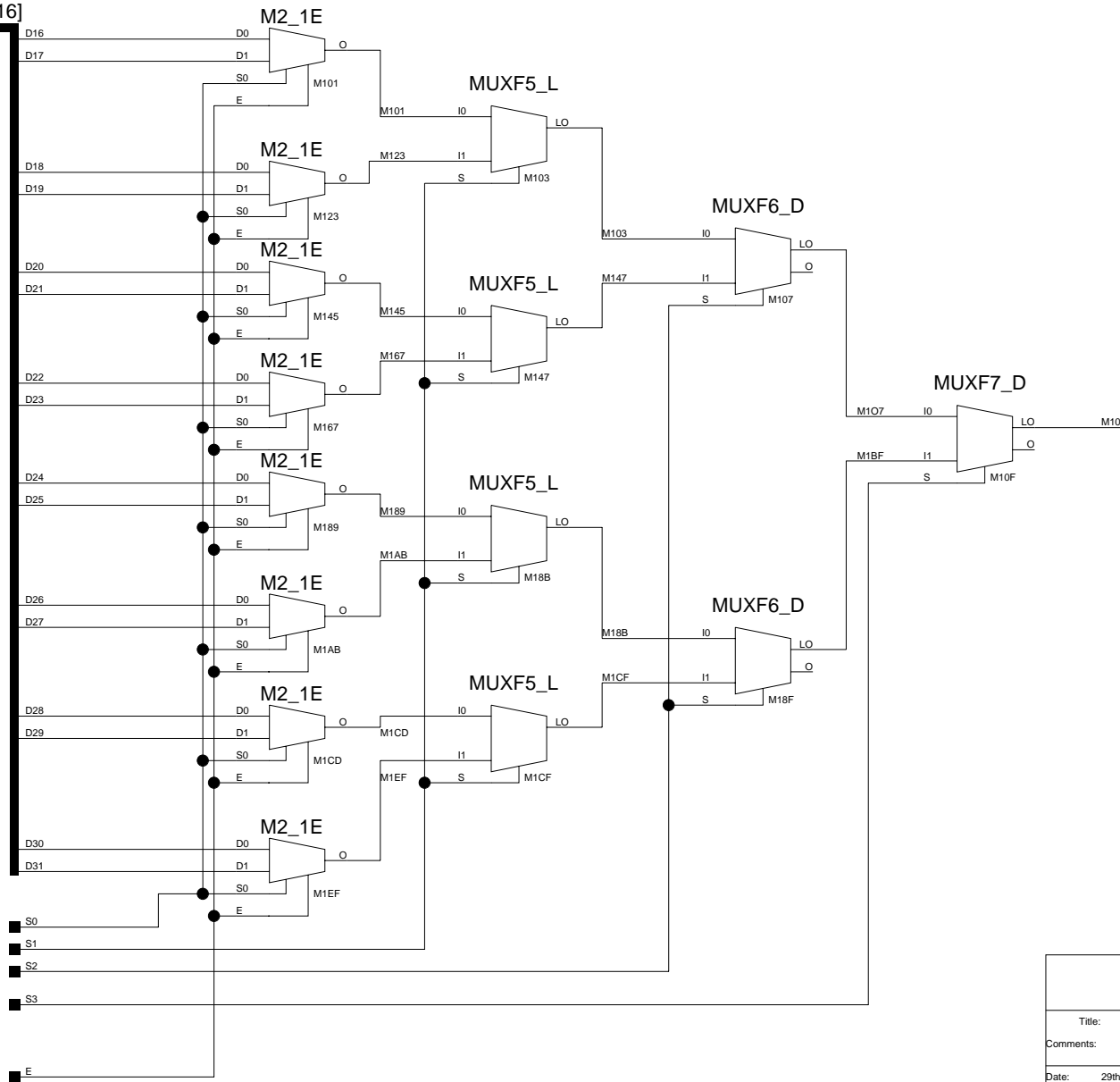
Implements as a DPRAM (8 slices)

drawn by KS  
Copyright (c) 1993, Xilinx Inc.

		JRG
Title: VIRTEx Family Fast M32E5bus Macro		
Comments: 32-to-1 bus Multiplexer with Enable		
Date: 29th September 2003	Ver: 1	
Sheet Size: B	Rev: A	



D[31:16]



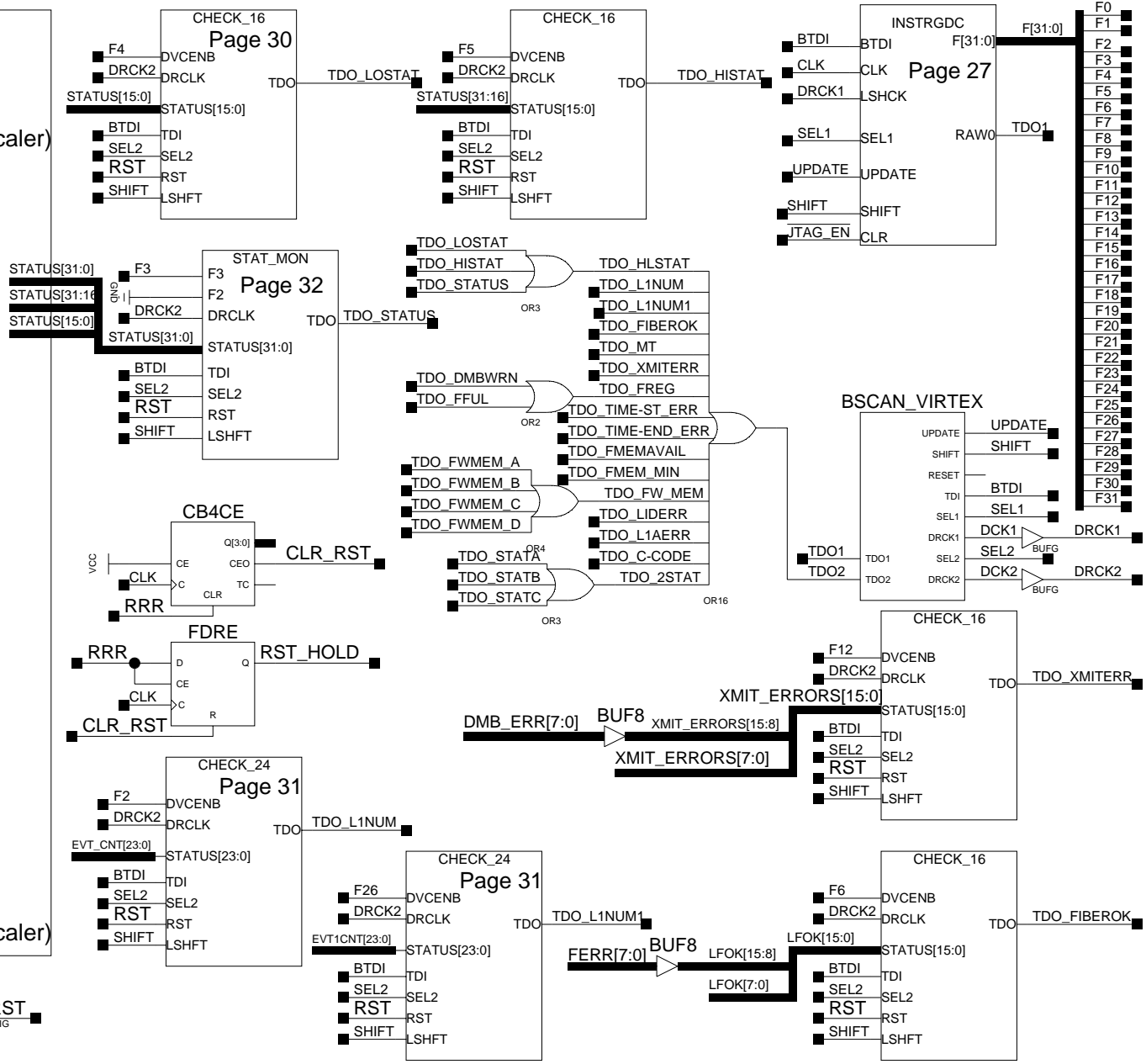
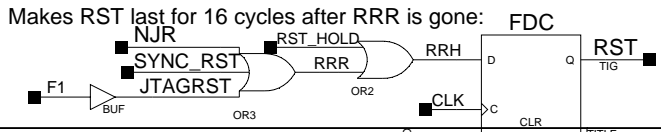
drawn by KS  
Copyright (c) 1993, Xilinx Inc.

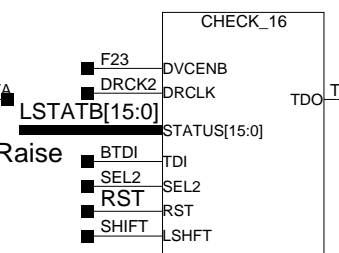
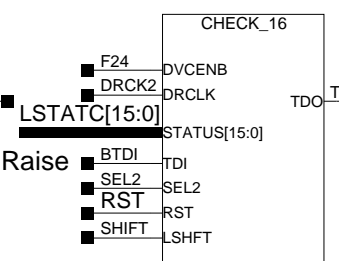
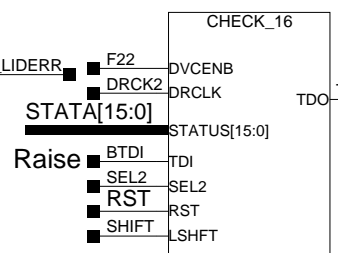
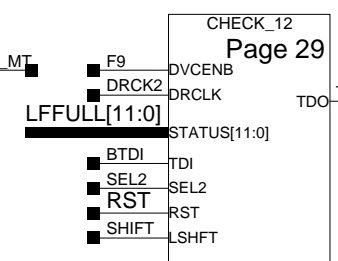
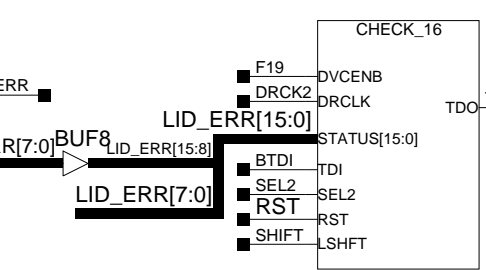
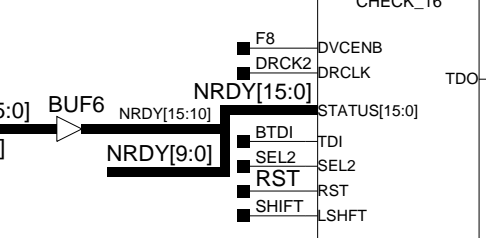
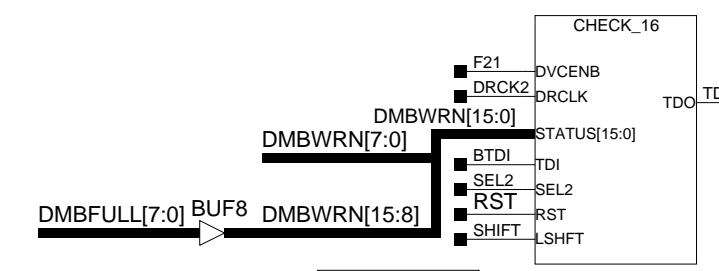
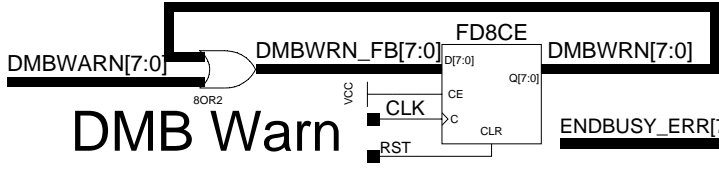
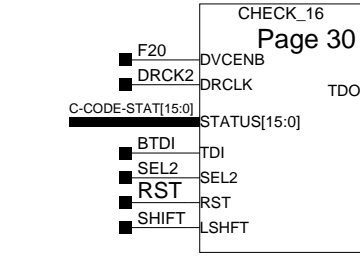
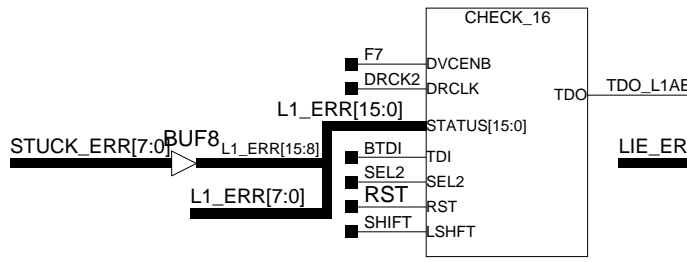
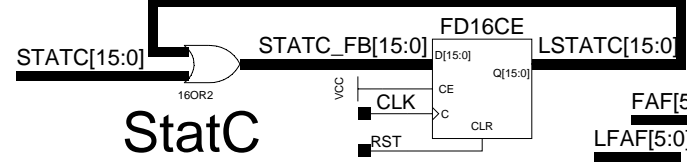
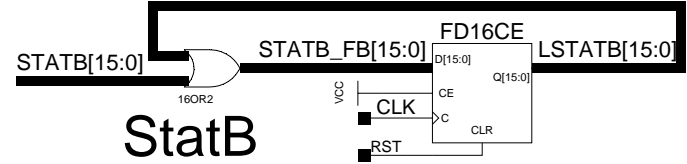
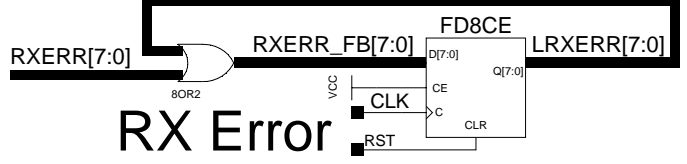
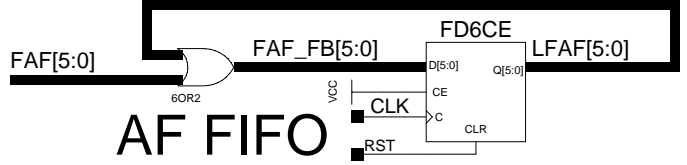
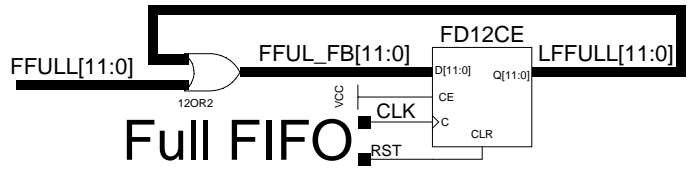


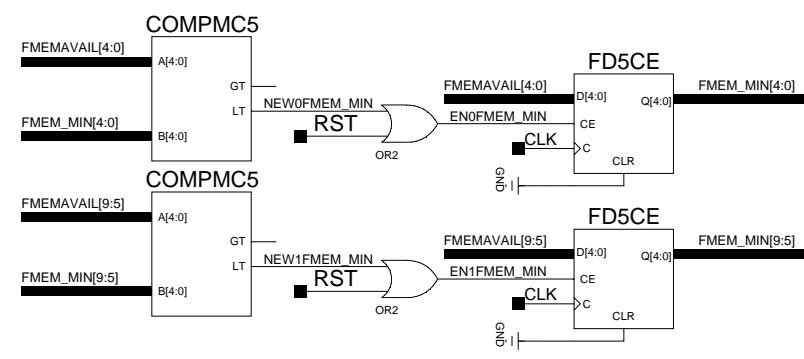
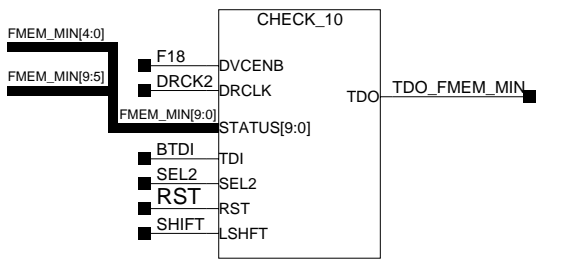
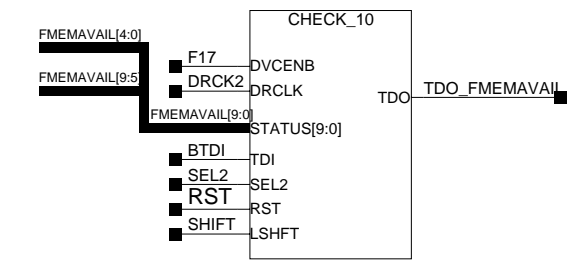
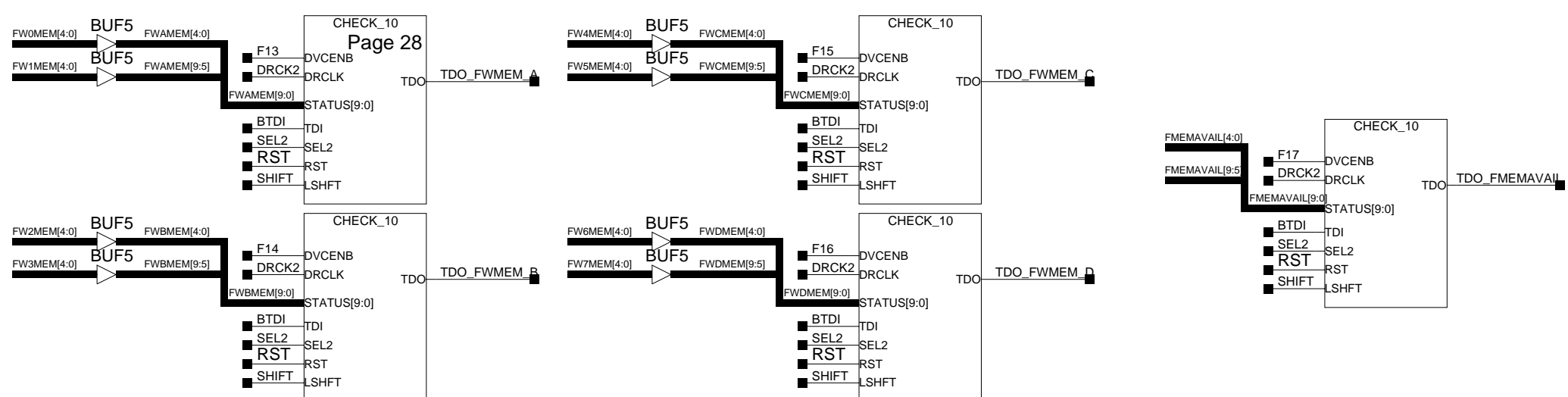
Title:	VIRTEX Family Fast M32E5bus Macro		
Comments:	32-to-1 bus Multiplexer with Enable		
Date:	29th September 2003	Ver:	1
Sheet Size:	B	Rev:	A

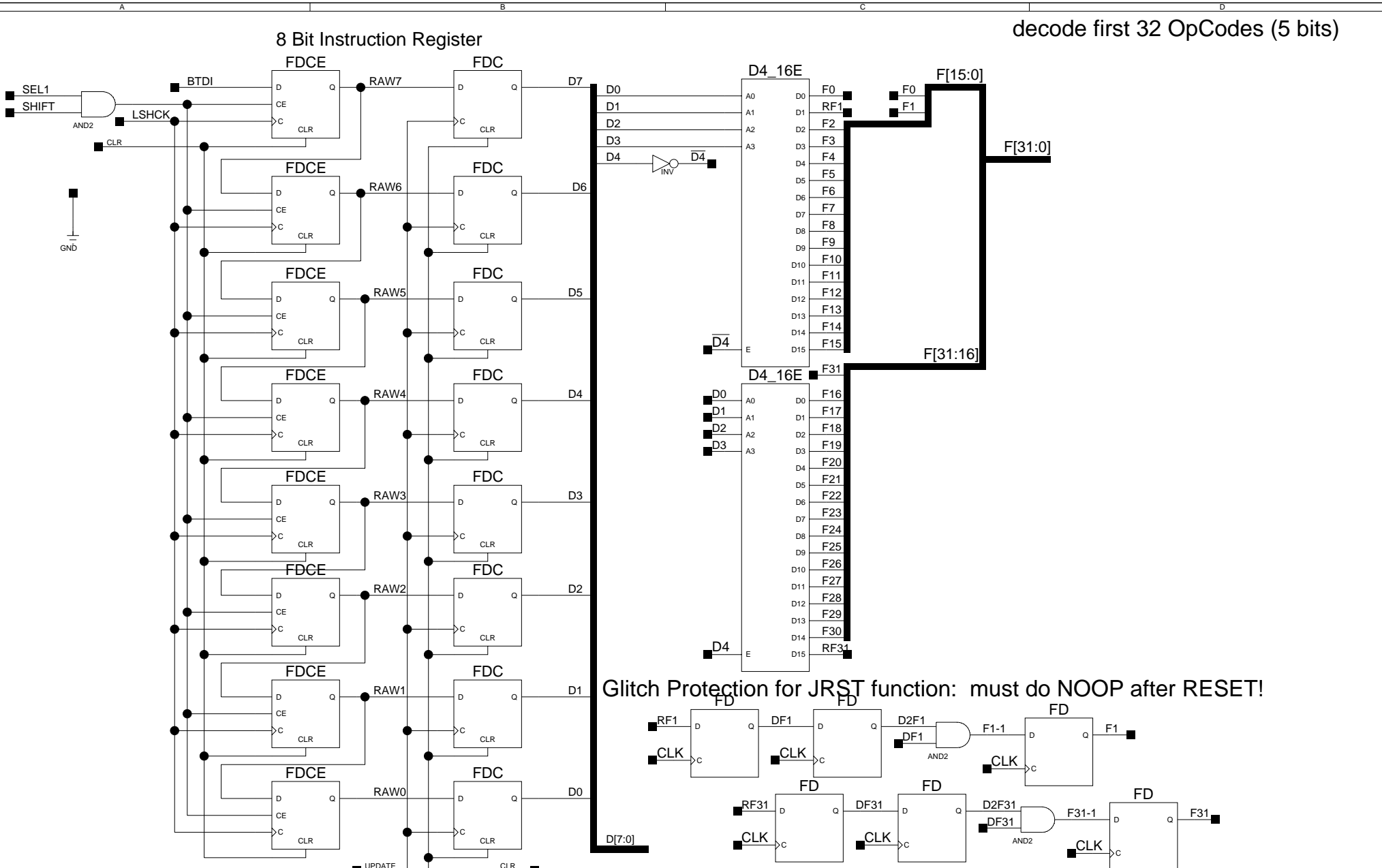
# JTAG Instruction Decode

OpCode	Function [OpName]
0	No Operation [NOOP]
1	FPGA Reset [toggle]
2	RdCtrl0 Current DDU L1A Number (24-bit scaler)
3	Check status (capture and shift) [32 bits]
4	Check status, low-word [16 bits]
5	Check status, high-word [16 bits]
7	6a Check FiberOK status (Check_FOK) [8 bits]
6	6b Check FiberErr status (Check_Ferr) [8 bits]
11	7a L1A Mismatch [8-bits]
26	7b Stuck Buffer Errors [8-bits]
25	8a FIFO Empty Status [10-bits]
20	8b Almost Full FIFOs [6-bits]
21	9 Full FIFOs [12-bits]
13	10a Timeout: start [8-bits]
17	10b RX Errors [8-bits]
14	11a Timeout: end-wait [8-bits]
15	11b Timeout: end-active [8-bits]
12	12a Data Xmit Errors [8-bits]
16	12b DMB Errors [8-bits]
13	Fiber 1 & 0 Write Memory [10-bits]
14	Fiber 3 & 2 Write Memory [10-bits]
15	Fiber 5 & 4 Write Memory [10-bits]
16	Fiber 7 & 6 Write Memory [10-bits]
17	FIFO Mem Available 1 & 0 [10-bits]
18	FIFO Mem Min Available 1 & 0 [10-bits]
19	*19a Lost In Data [8-bits]
18	*19b Lost In Event [8-bits]
20	C-Code Status [16-bits]
21	DMB Warn/Full Status [16-bits]
22	Status Register A [16-bits]
23	Status Register B [16-bits]
24	Status Register C [16-bits]
26	RdCtrl1 Current DDU L1A Number (24-bit scaler)





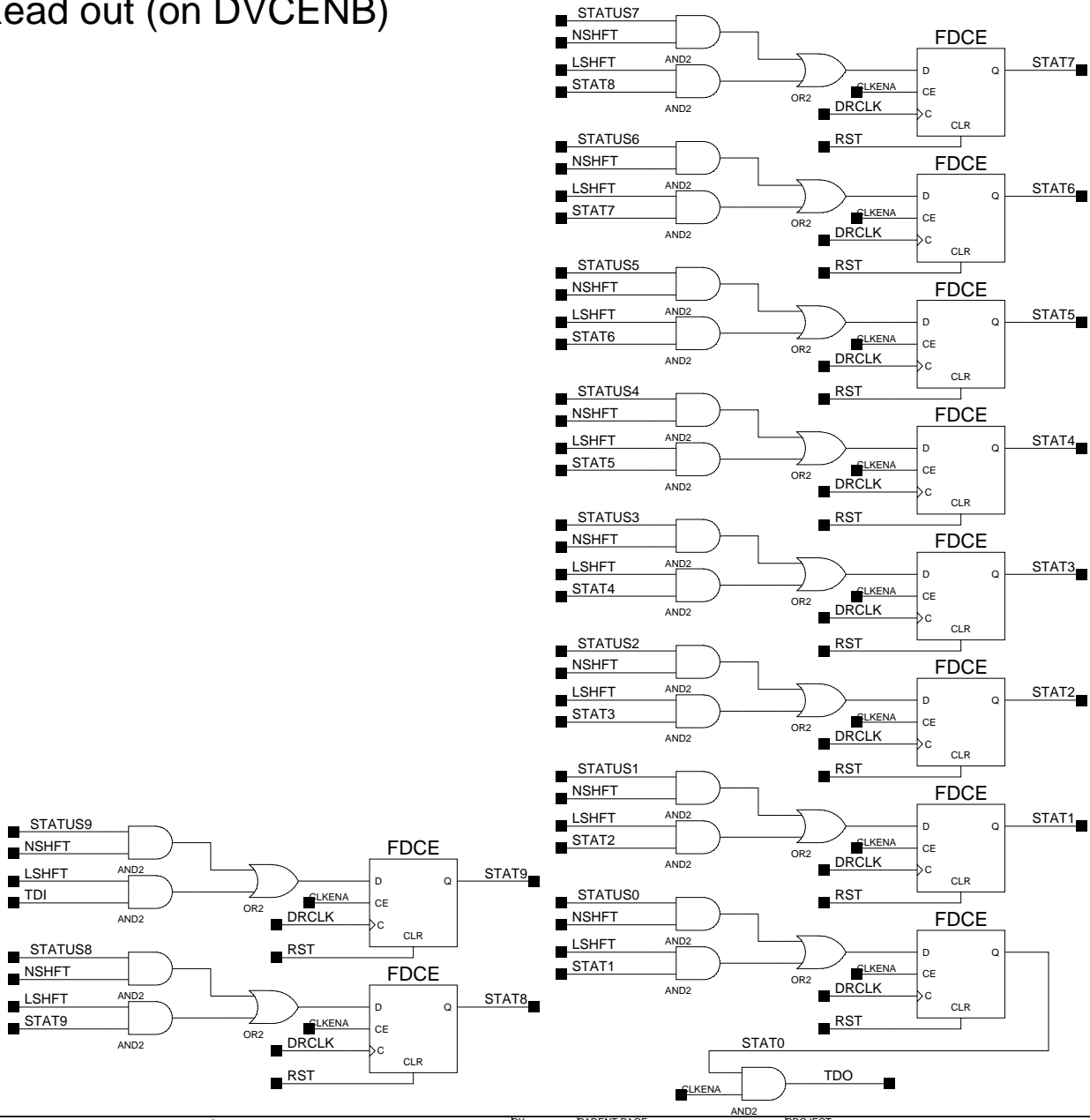
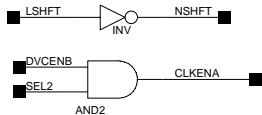




# 10-bit JTAG Register Read out (on DVCENB)

STATUS[9:0]

- STATUS9
- STATUS8
- STATUS7
- STATUS6
- STATUS5
- STATUS4
- STATUS3
- STATUS2
- STATUS1
- STATUS0



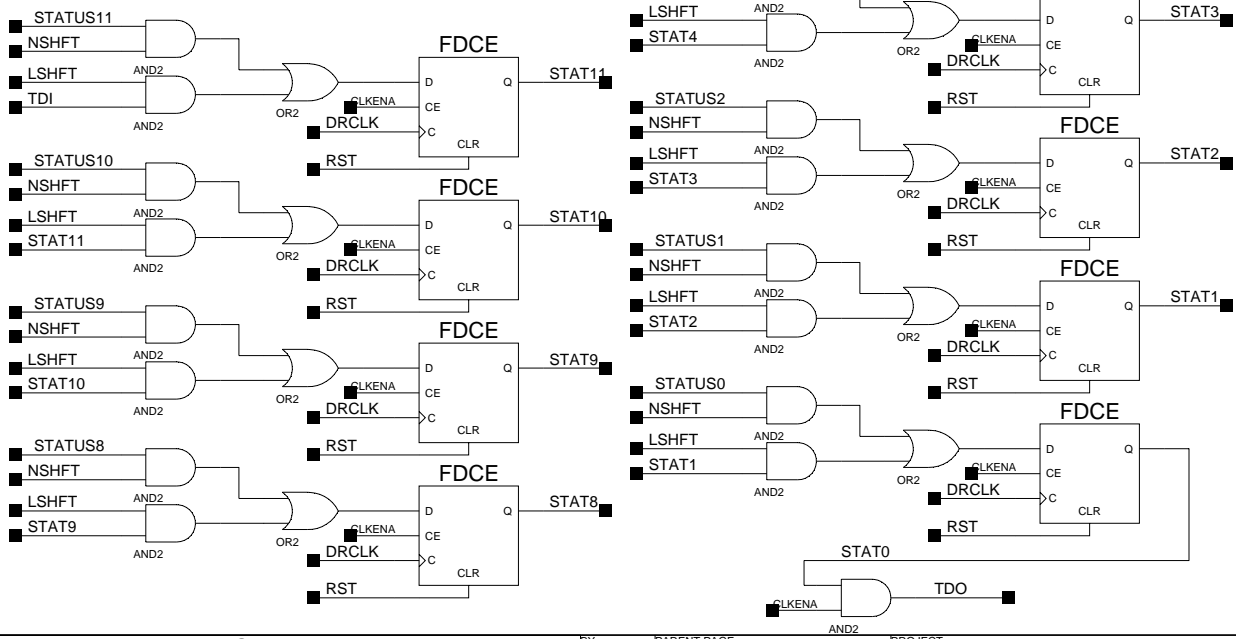
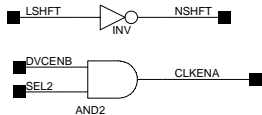
TITLE

PROJECT

# 12-bit JTAG Register Read out (on DVCENB)

STATUS[11:0]

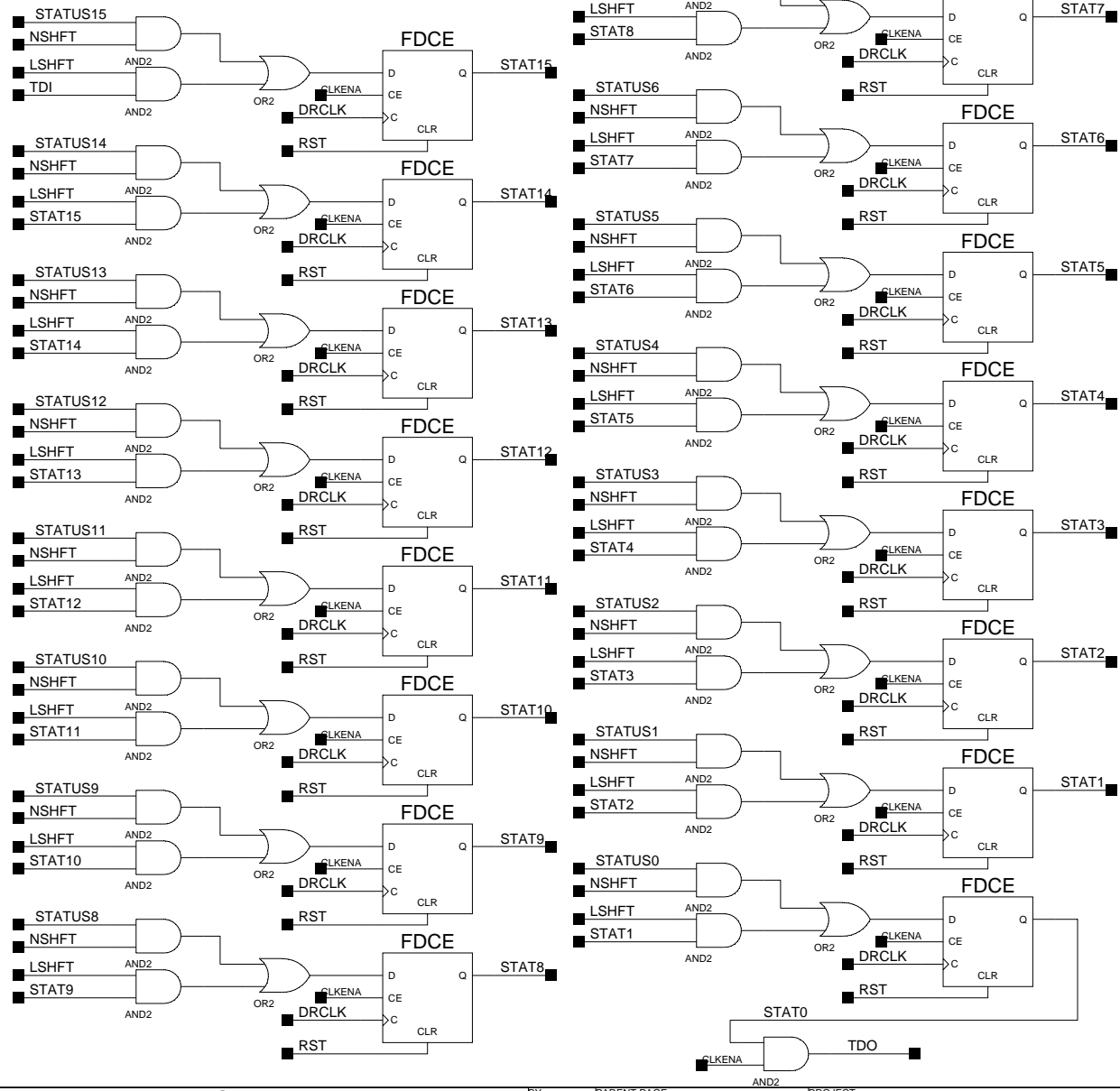
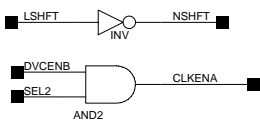
- STATUS11
- STATUS10
- STATUS9
- STATUS8
- STATUS7
- STATUS6
- STATUS5
- STATUS4
- STATUS3
- STATUS2
- STATUS1
- STATUS0



# 16-bit JTAG Register Read out (on DVCENB)

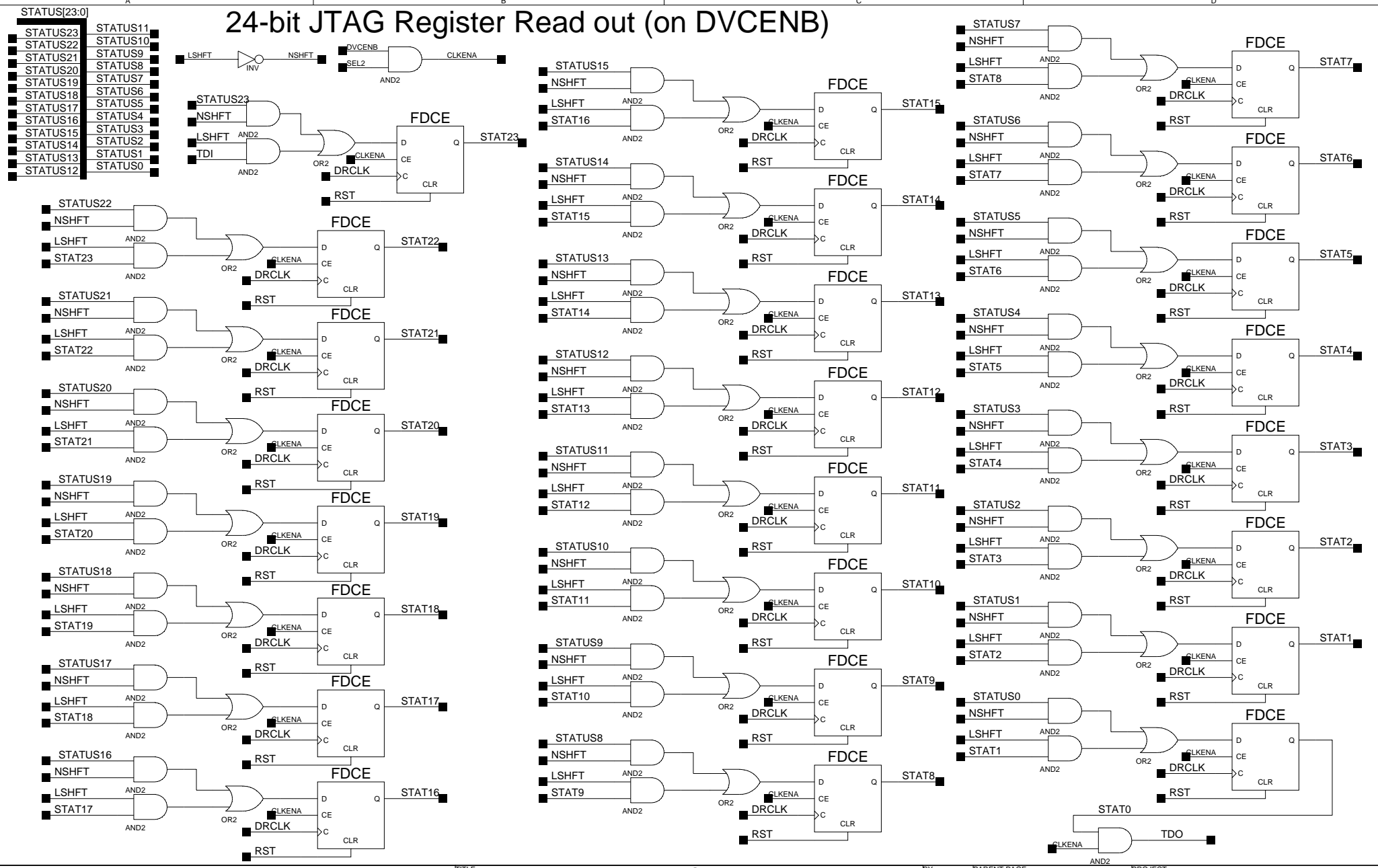
STATUS[15:0]

- STATUS15
- STATUS14
- STATUS13
- STATUS12
- STATUS11
- STATUS10
- STATUS9
- STATUS8
- STATUS7
- STATUS6
- STATUS5
- STATUS4
- STATUS3
- STATUS2
- STATUS1
- STATUS0

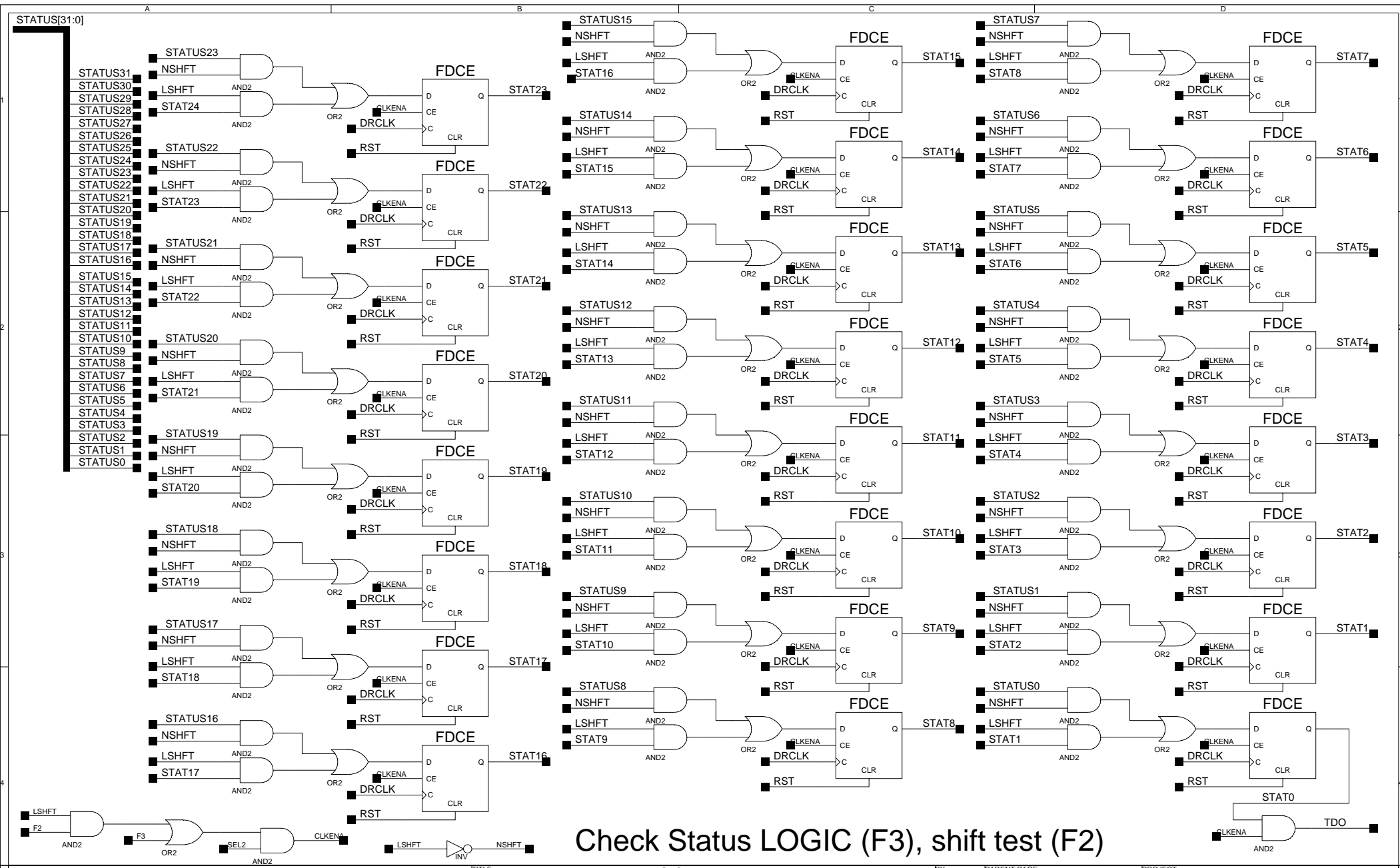




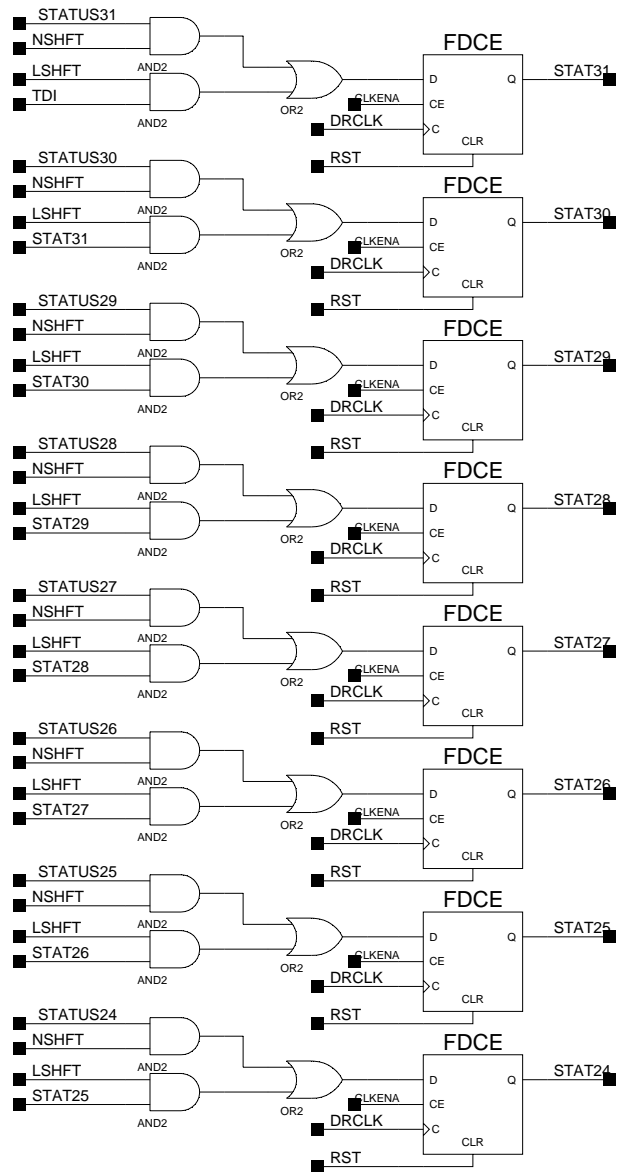
# 24-bit JTAG Register Read out (on DVCENB)



STATUS23	STATUS11
STATUS22	STATUS10
STATUS21	STATUS9
STATUS20	STATUS8
STATUS19	STATUS7
STATUS18	STATUS6
STATUS17	STATUS5
STATUS16	STATUS4
STATUS15	STATUS3
STATUS14	STATUS2
STATUS13	STATUS1
STATUS12	STATUS0



Check Status LOGIC (F3), shift test (F2)

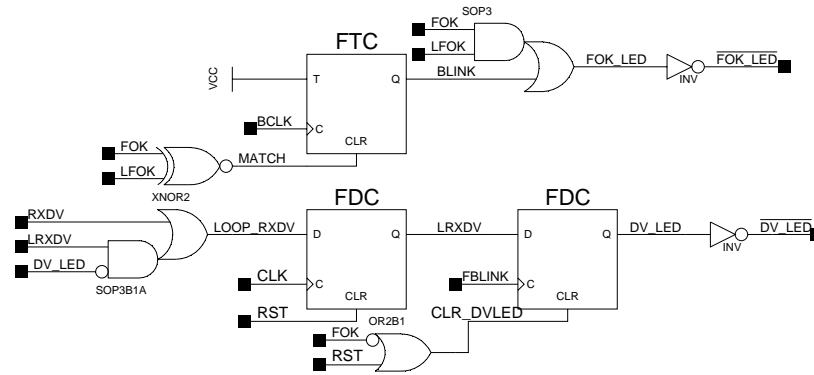


# FOK LED

- LIT == Link is alive and well
- BLINK == Link not ready
- OFF == Link not present

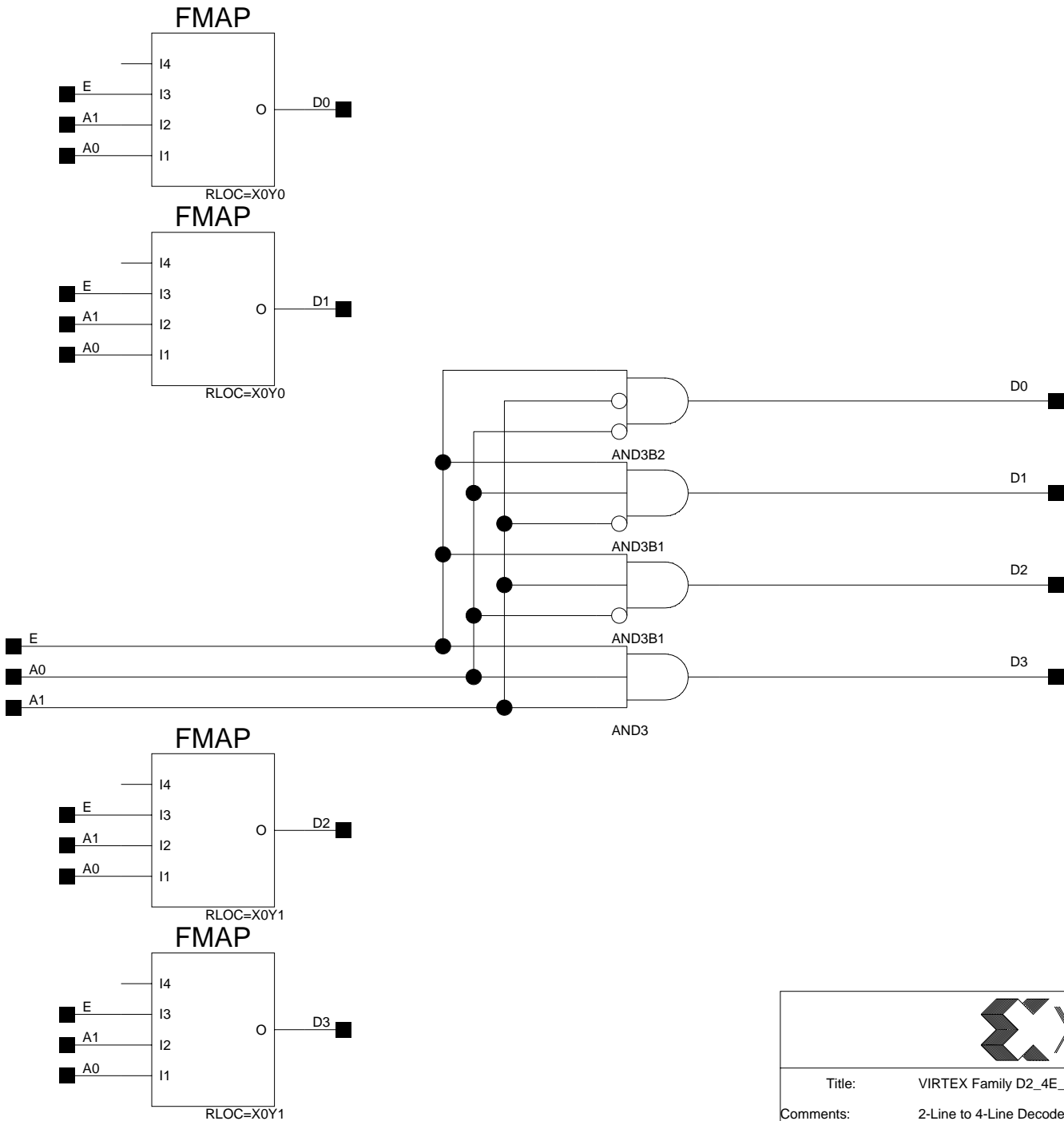
# DAV LED

- LIT == Active Data Xmit
- OFF == No data to Xmit



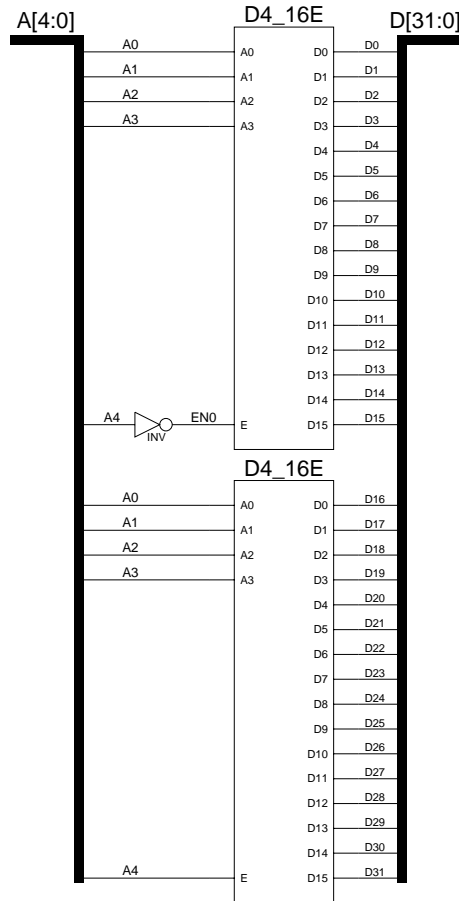
JRG

Title:	FIBERLED	
Comments:	Custom LED Slow-Blink Control for Fiber Inputs	
Date:	27th January 2004	Ver: 1
Sheet Size: B		Rev: A



JRG

Title:	VIRTEX Family D2_4E_rpm Macro		
Comments:	2-Line to 4-Line Decoder/ Demultiplexer w/ Enable, RPM		
Date:	11th November 2003	Ver:	1
Sheet Size:	A	Rev:	A



Title: VIRTEX Family Decode 32E Macro		<b>JRG</b>
Comments: 5-bit to 32-bit Decoder		
Date: 30th September 2003	Ver: 1	
Sheet Size: B	Rev: A	

drawn by KS  
Copyright (c) 1993, Xilinx Inc.

RPM\_GRID=GRID



D[17:0]

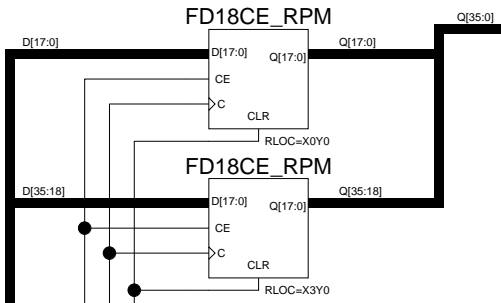
- CE
- C
- CLR



Title:	VIRTEX Family FD18CE_RPM Macro	
Comments:	18-Bit Data Register w/ Clock Enable & Asynchronous Clr, GRID Coords X0Y0-X0Y8	
Date:	29th October 2003	Ver: 1
Sheet Size:	B	Rev: A

JRG

drawn by KS  
Copyright (c) 1993, Xilinx Inc.



D[35:0]

■ CE

■ C

■ CLR



Title: VIRTEX Family FD36CE\_RPM Macro

Comments: 36-Bit Data Register w/  
Clock Enable & Asynchronous Clr, GRID Coords X0Y0-X3Y8

Date: 30th October 2003

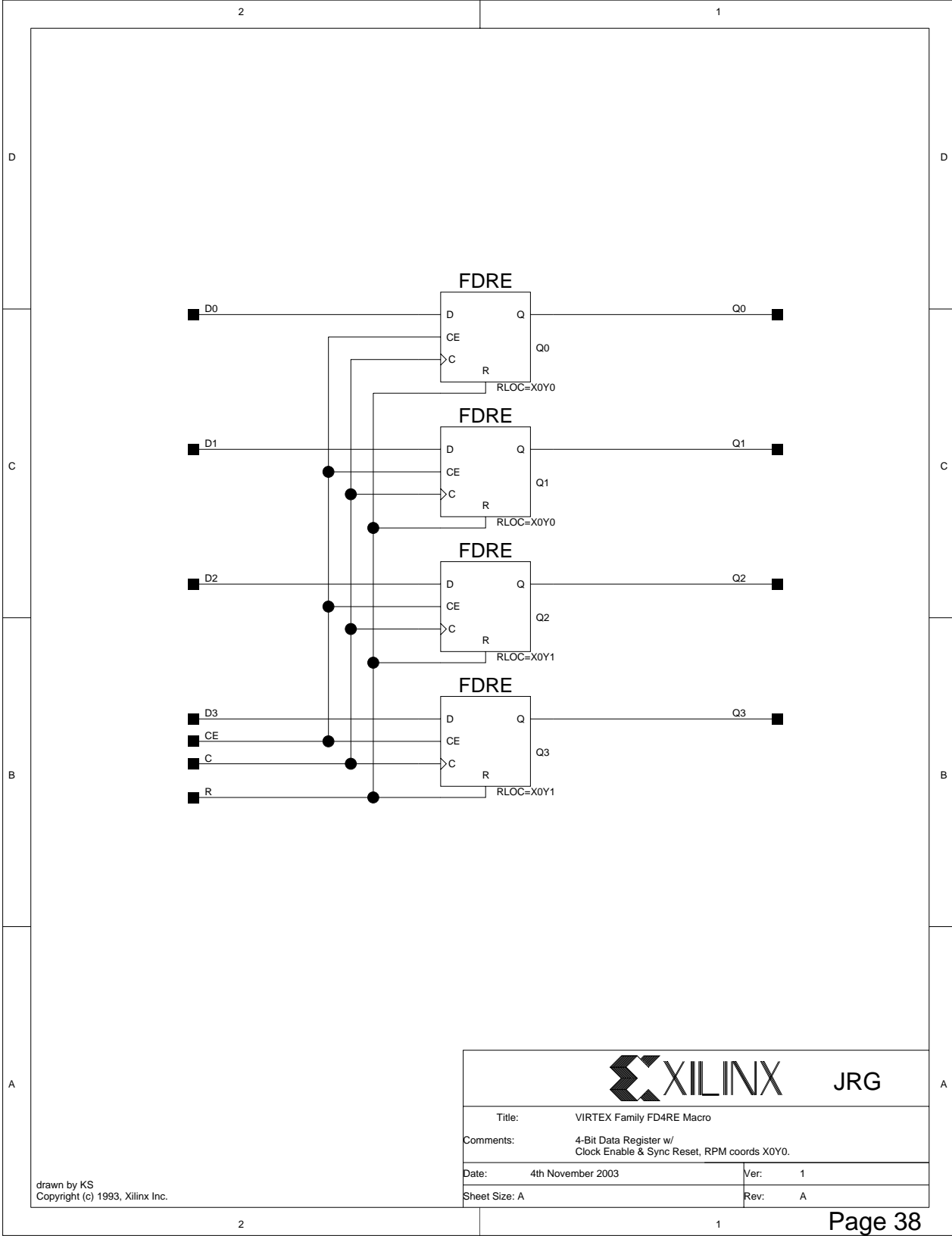
Ver: 1

Sheet Size: B


Rev: A

JRG



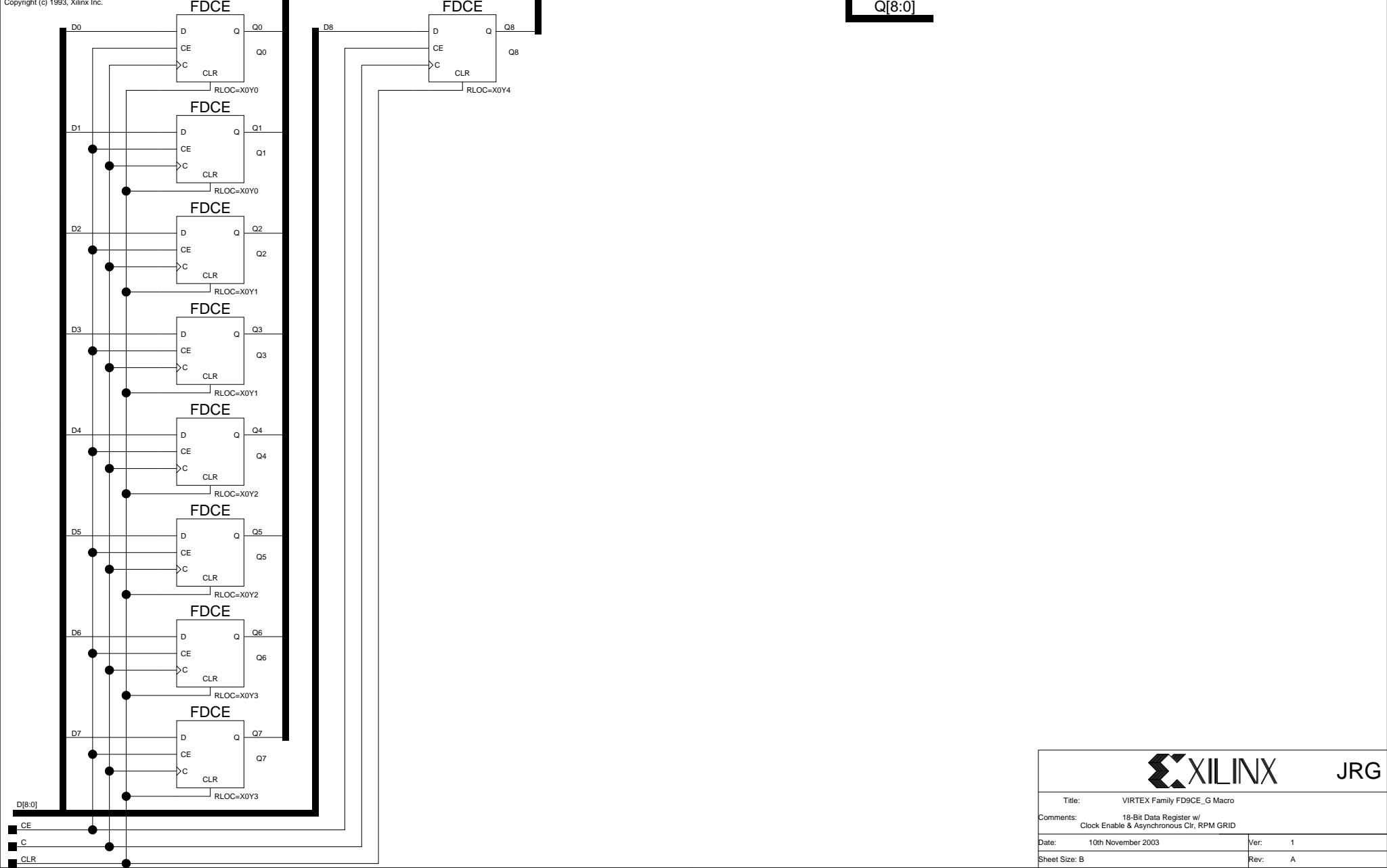



drawn by KS  
Copyright (c) 1993, Xilinx Inc.

		JRG
Title: VIRTEX Family FD4RE Macro		
Comments: 4-Bit Data Register w/ Clock Enable & Sync Reset, RPM coords X0Y0.		
Date: 4th November 2003	Ver: 1	
Sheet Size: A	Rev: A	

drawn by KS  
Copyright (c) 1993, Xilinx Inc.

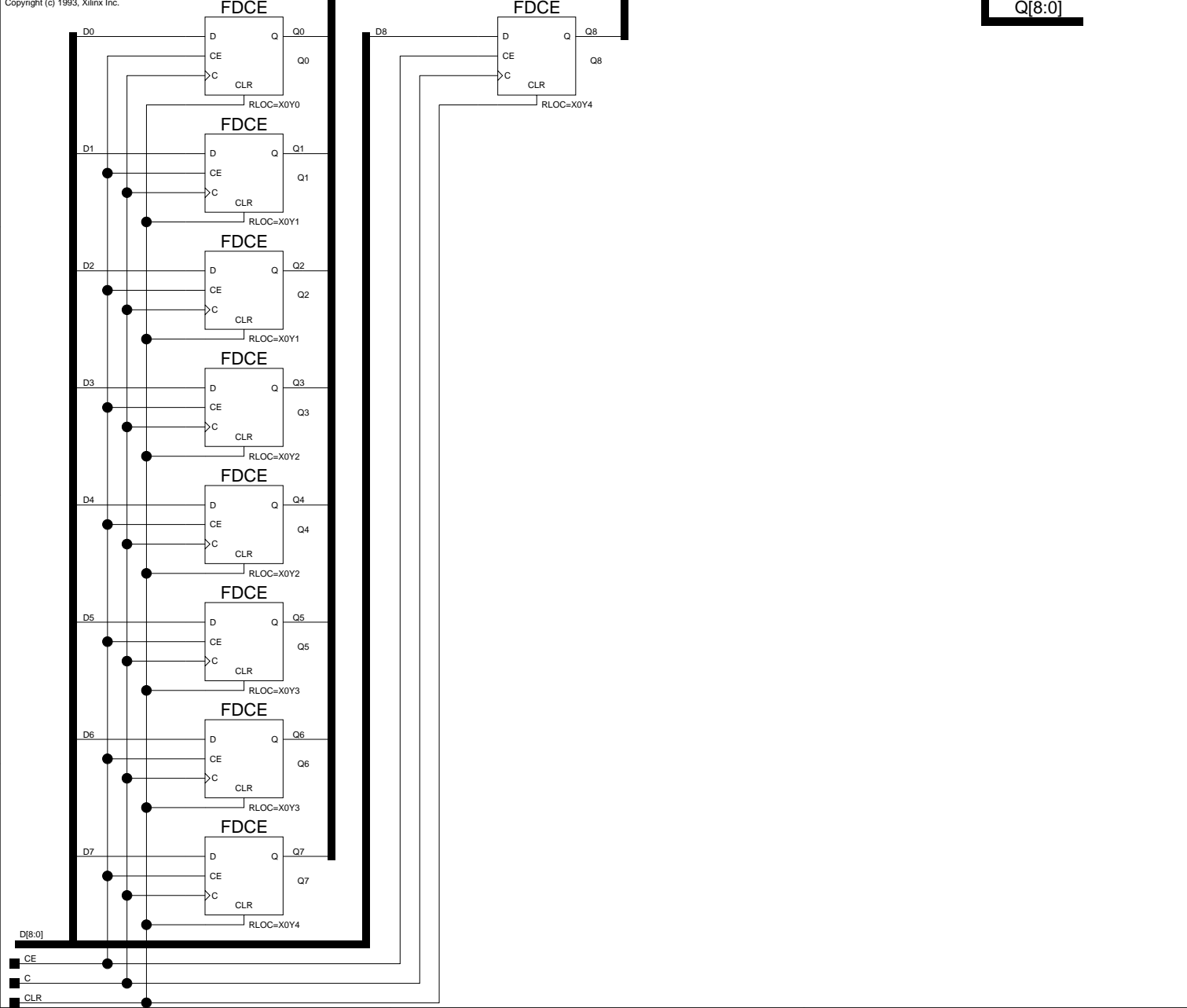
RPM\_GRID=GRID



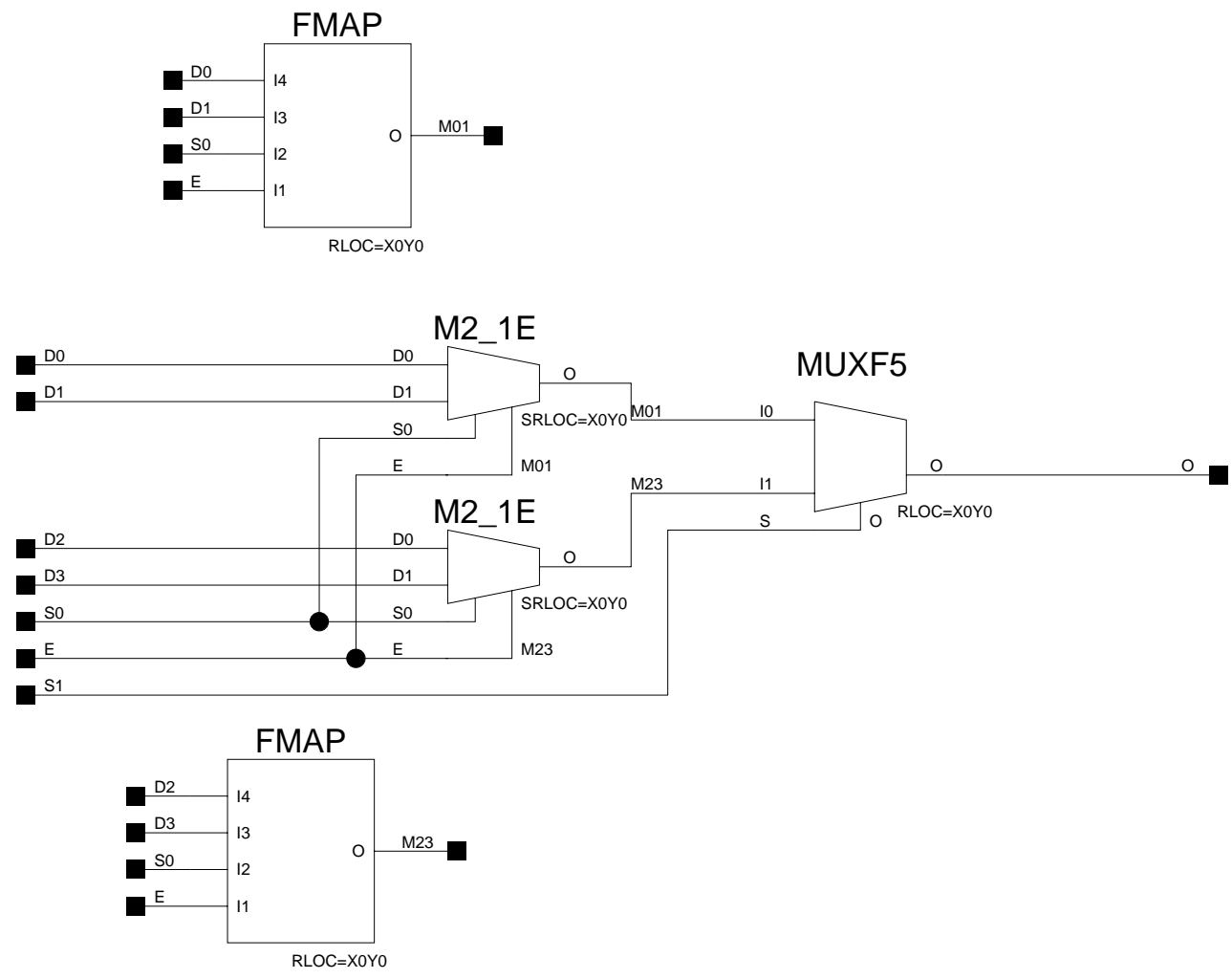
		<b>JRG</b>
Title: VIRTEX Family FD9CE_G Macro		
Comments: 18-Bit Data Register w/ Clock Enable & Asynchronous Clr, RPM GRID		
Date: 10th November 2003	Ver: 1	
Sheet Size: B	Rev: A	

drawn by KS  
Copyright (c) 1993, Xilinx Inc.

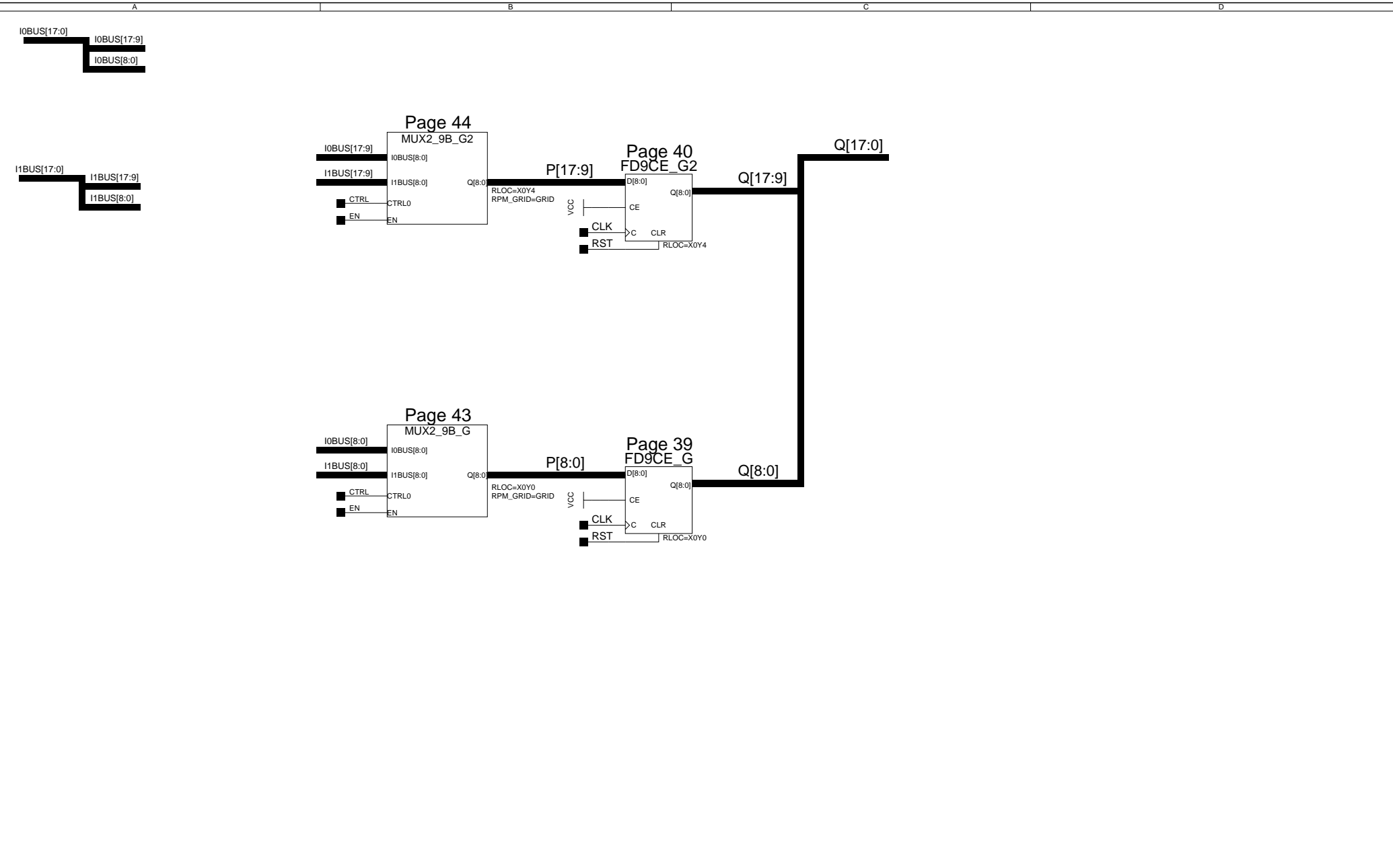
RPM\_GRID=GRID

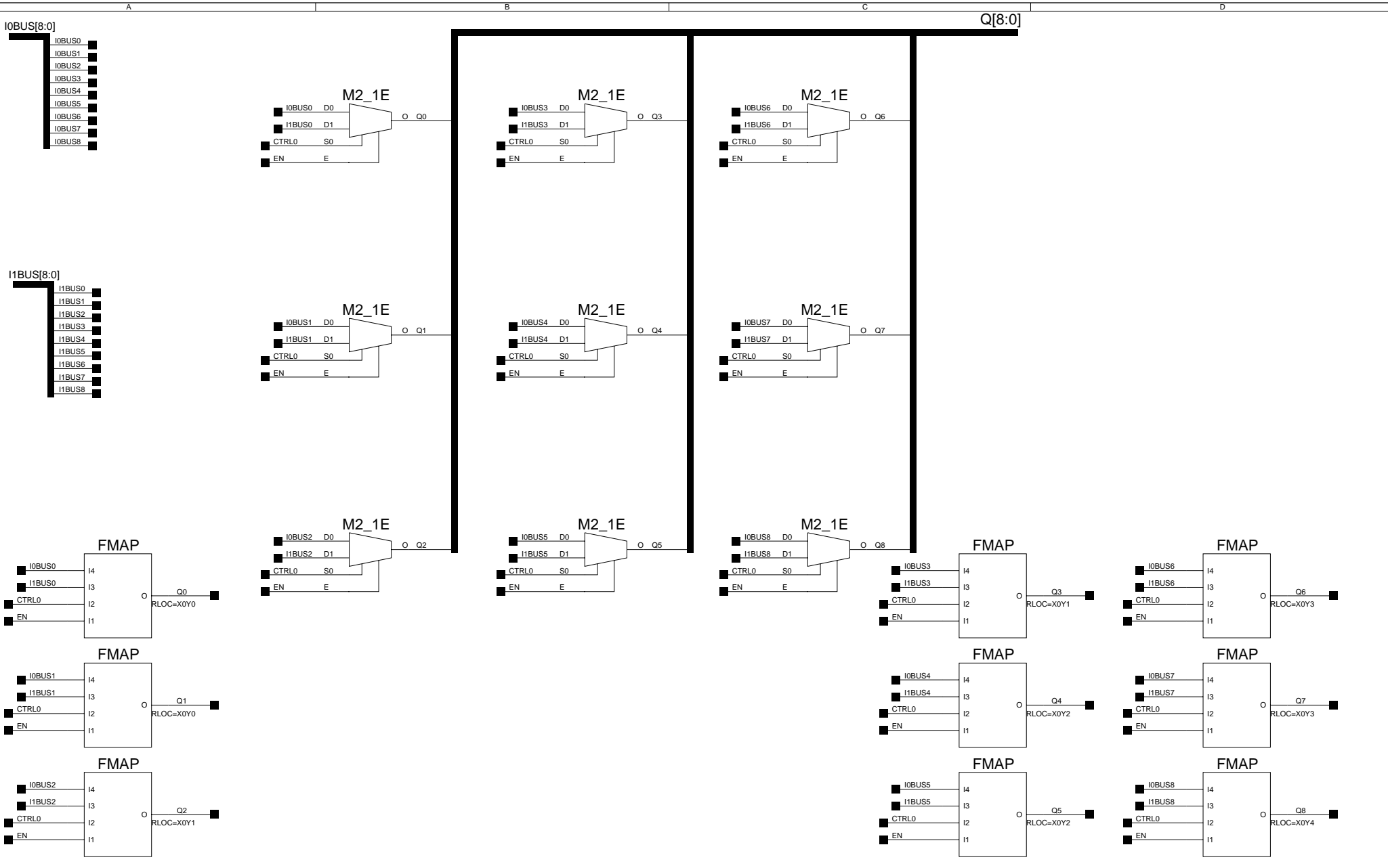


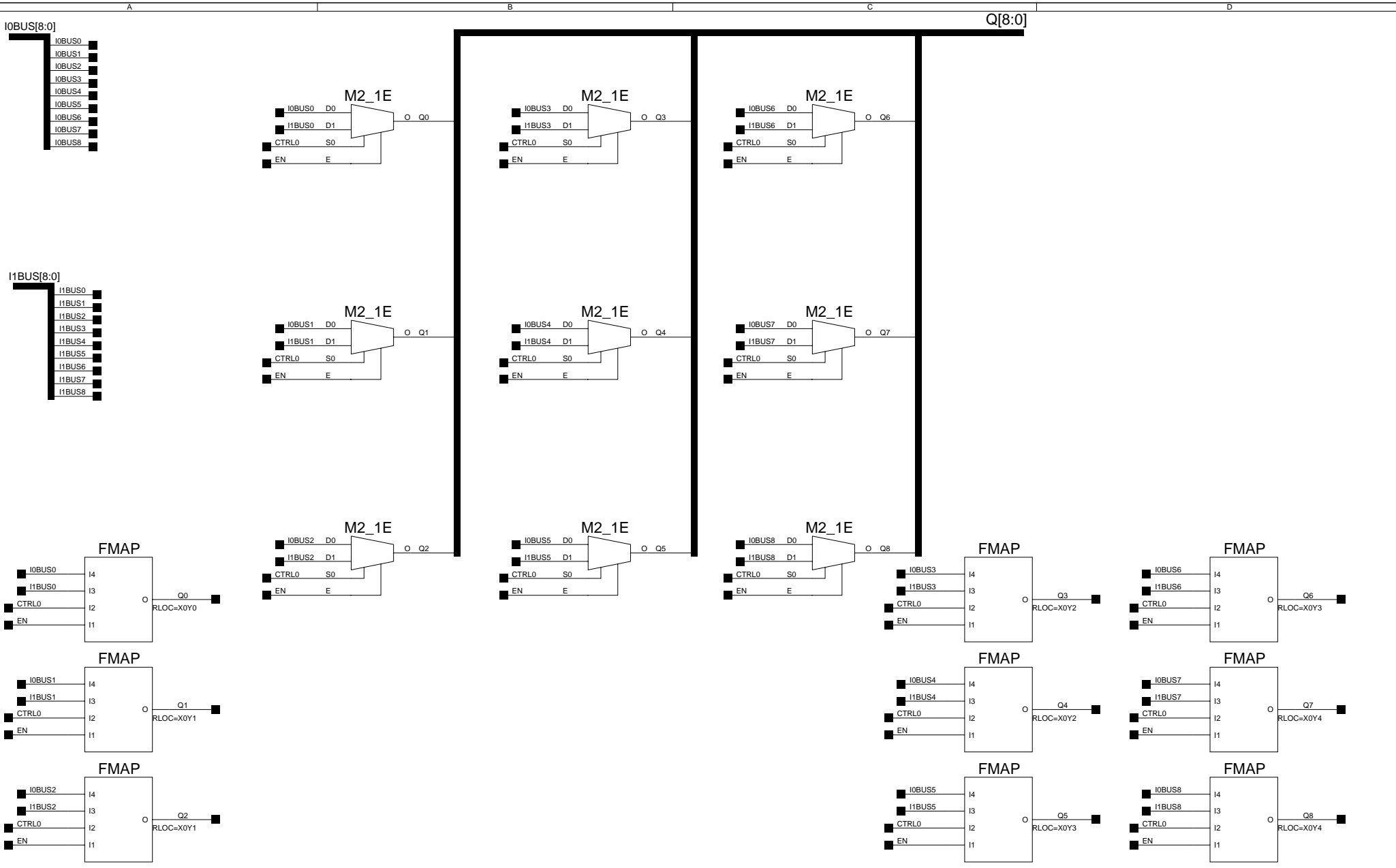
		<b>JRG</b>
Title: VIRTEX Family FD9CE_G2 Macro		
Comments: 18-Bit Data Register w/ Clock Enable & Asynchronous Clr, RPM GRID		
Date: 10th November 2003	Ver: 1	
Sheet Size: B	Rev: A	

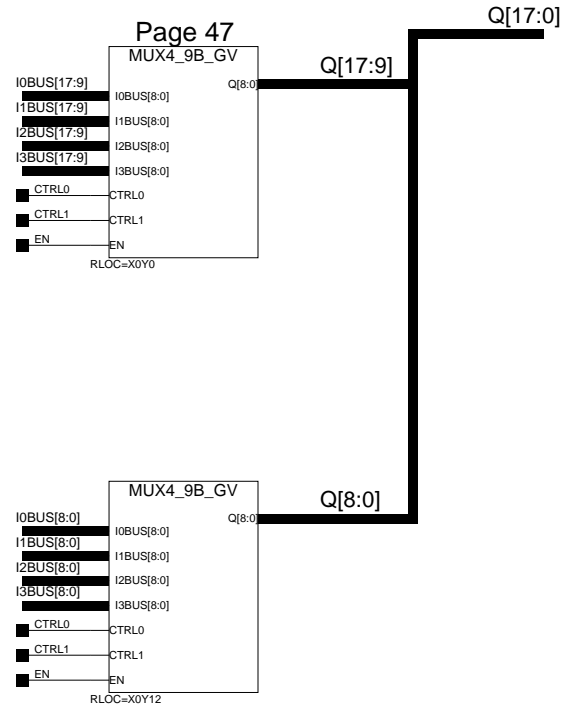
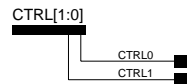
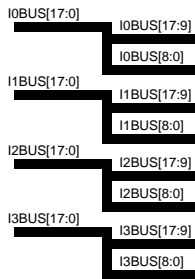


Title:	VIRTEX Family M4_1E_RPM Macro		
Comments:	4-to-1 Multiplexer with Enable		
Date:	31st October 2003	Ver:	1
Sheet Size:	A	Rev:	B

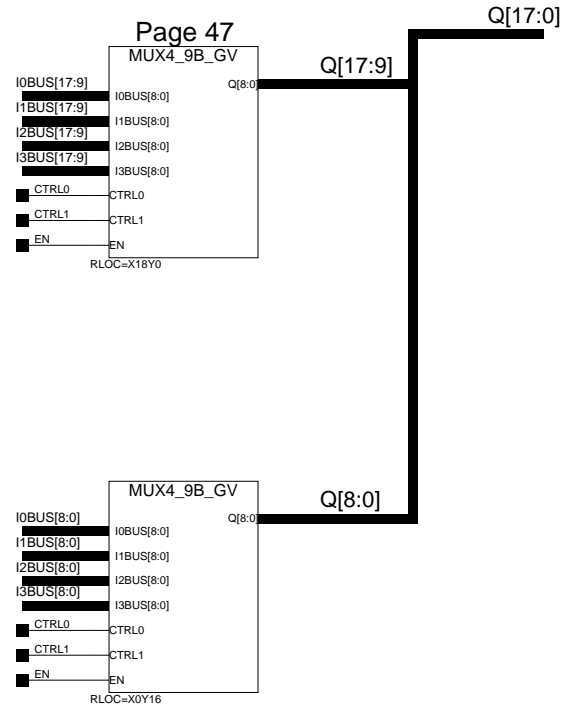
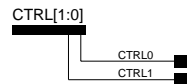
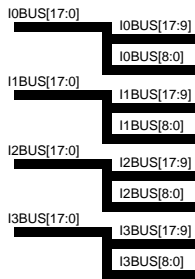


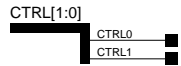
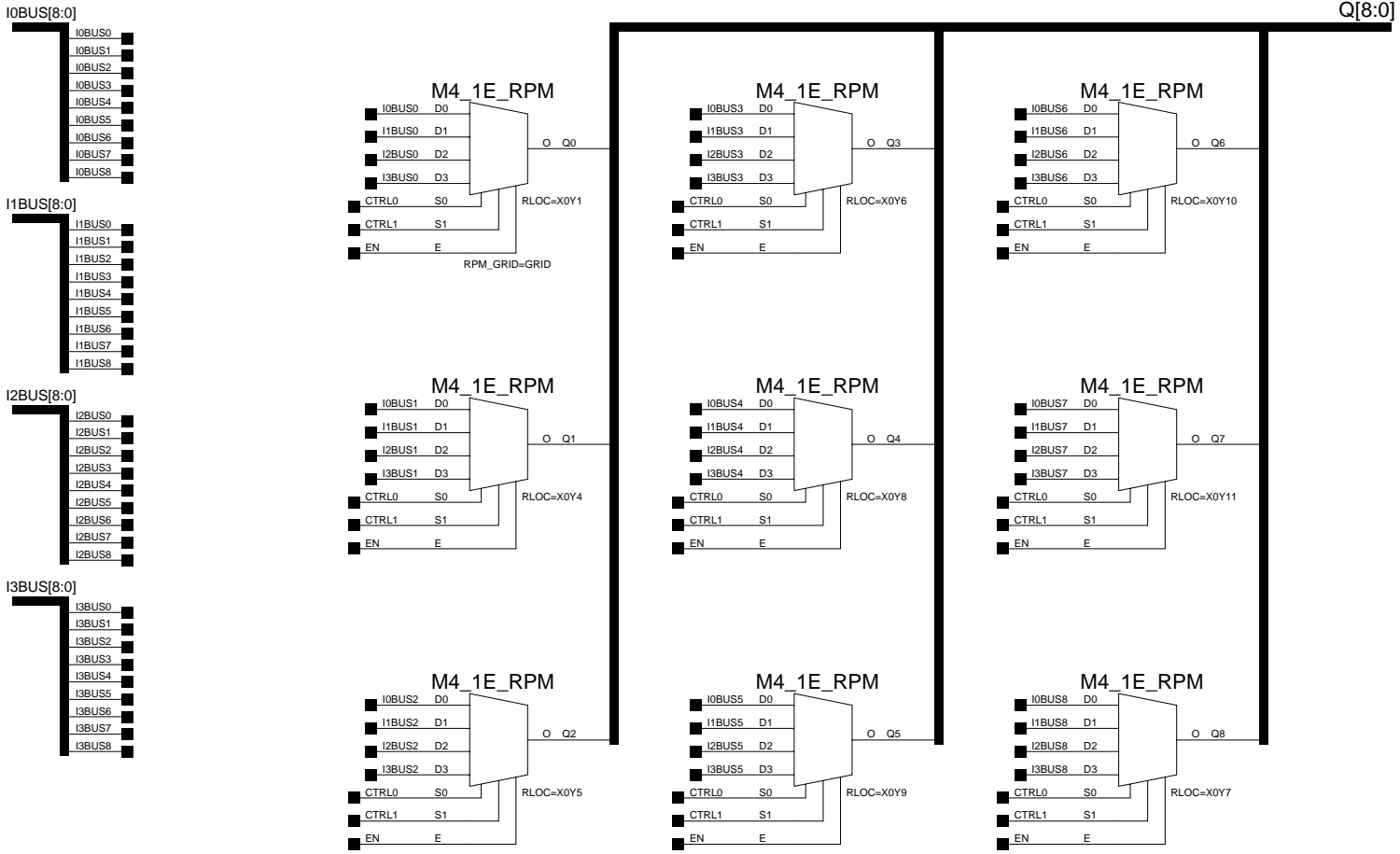




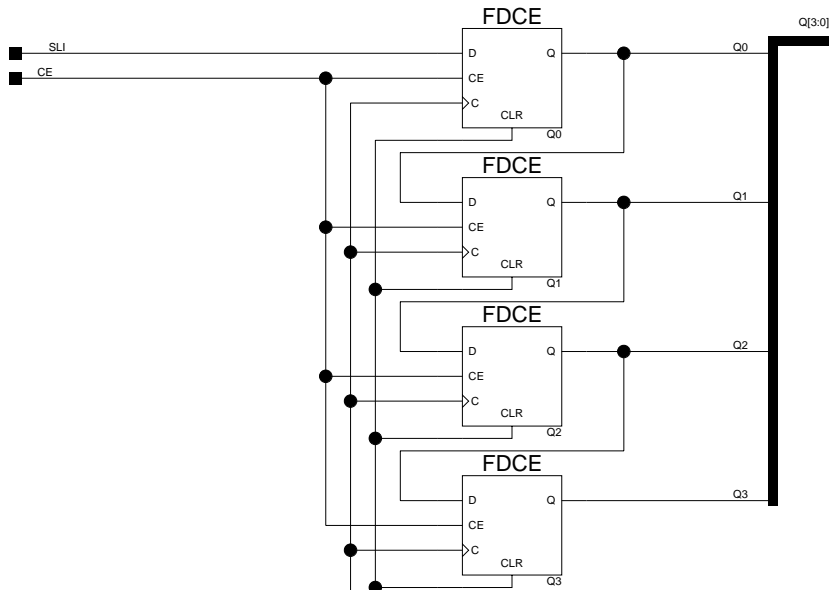








drawn by KS  
Copyright (c) 1993, Xilinx Inc.



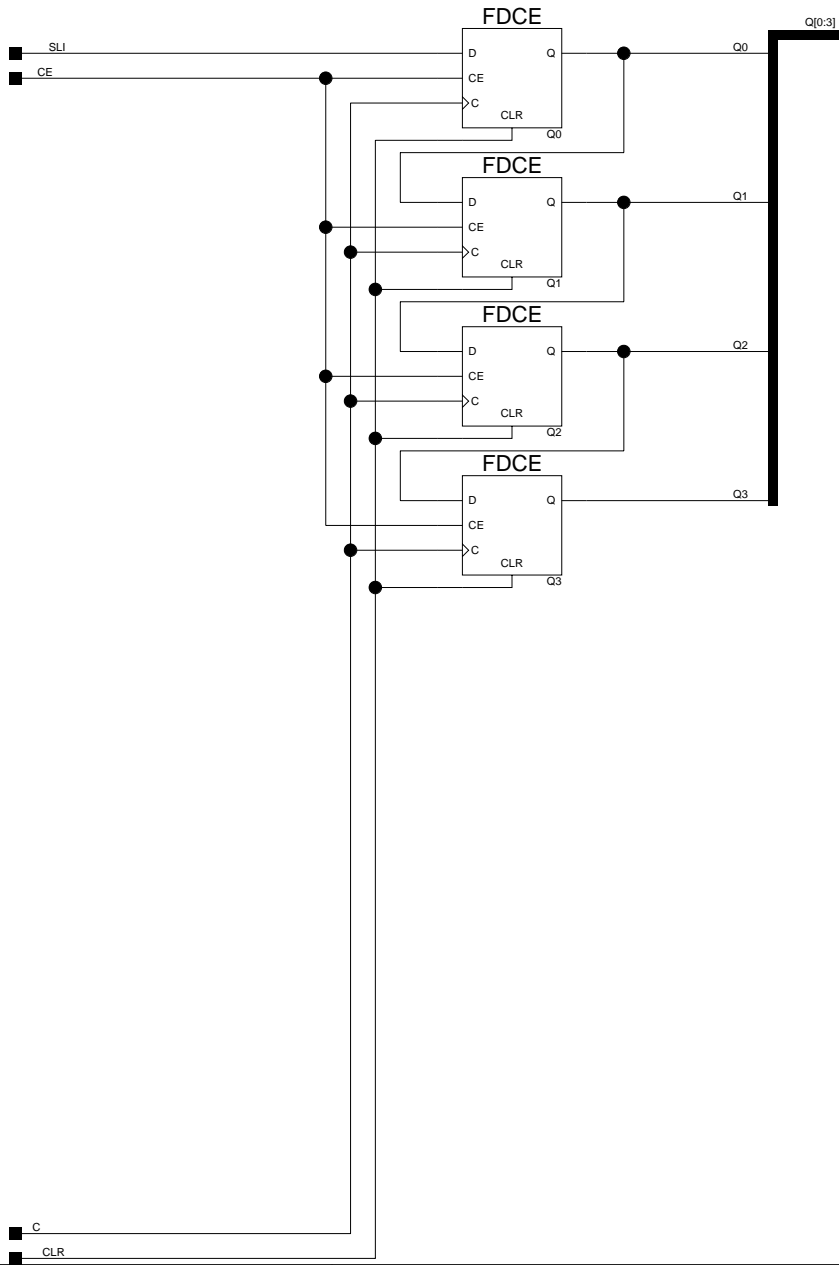
C  
CLR



JRG

Title: VIRTEX Family SR4CE Macro	
Comments: 4-bit Serial-In Parallel-Out Shift Register w/ Enable and Async Clr	
Date: 3rd October 2003	Ver: 1
Sheet Size: B	Rev: A

drawn by KS  
Copyright (c) 1993, Xilinx Inc.



Title: VIRTEX Family SR4CE_R Macro		JRG
Comments: 4-bit Serial-In Parallel-Out, REVERSED ORDER on OUTPUT BUS! Shift Register w/ Enable and Async Clr		
Date: 3rd October 2003	Ver: 1	
Sheet Size: B	Rev: A	