

# In5Ctrl (file Oddu\_in) 9-25-2006\_11:20

## CMS CSC DDU5, Input Control FPGAs DF023A01 Version 23

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 FakeLen 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)  
 -- could use code "8" instead...  
 - Should we Reset RxErr Monitor? How? v12: fix FF0/1, retune MEM\_CTRL, add In\_C-CodeStat JTAG Reg (F20)  
 v13: remove RxErr & NoLiveFibers from SingleErr, add RdCtrl-1 L1\_Count JTAG Reg (F26)  
 v14: tune Mon.Reg.Resets, add DMB Warn/Full monitor & register (f21), 2 I/O for IDMB\_Full to DDUctrl  
 v15: Tune End\_Timeout logic--sends extra C-code with LW signal  
 v16: add diagnostic traps on F25 & 28, uses RHL  
 v17: tune InUnit WrNext for WE+RE case, tune f6/7error  
 v18: tune DMBerr/SCAfull register timing, add ML1Err & SingWarn to C-Code, tune SyncErr/iDMBfull  
 v19: bring FILLED to SingErr, added Mode7/LEDm8+ Mux's  
 v20: fix DMBfull-to-JTAG monitor logic  
 v21: add 2\*32 bit Fiber Memory diagnostic (F30,31)  
 v22: replace F\_OK with LFOK for InUnits;  
 v23, RxErr sets SingleWarn, but LRxErr goes on Stat1  
 7: Set Fake\_L1A (data passthrough) filled on Stat30, LDLLerr on STAT31  
 8: Show STAT31-0 on LAs, ~FPGA version on LEDs

**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



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

DDU5in\In5Ctrl\in5ctrl

\* af\_clb\_5x31rpm has Core EDN file  
 PROGRAM takes < 55ms

VME Broadcast Addresses:  
 24=OSU-TCB "Test Control Board"  
 25=DMB  
 26=TMB  
 27=Both DMB and TMB  
 28=DDU  
 22=DCC

PromID: 05026093h  
 FPGAid: 31266093h

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

- To Do:
- COMPARE LINUM & 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.
  - Add InUnit Check for DMB Timeouts & set FMM ErrOut

Special Startup Order:

- 2) Release WE
- 3) Release DLL
- 5) En. Outputs
- 6) DONE

DDU Format Since DDUctrl v15:

H1: 0x/51/NN.NNNN/XXX/1.H/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

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^^ GBE\_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

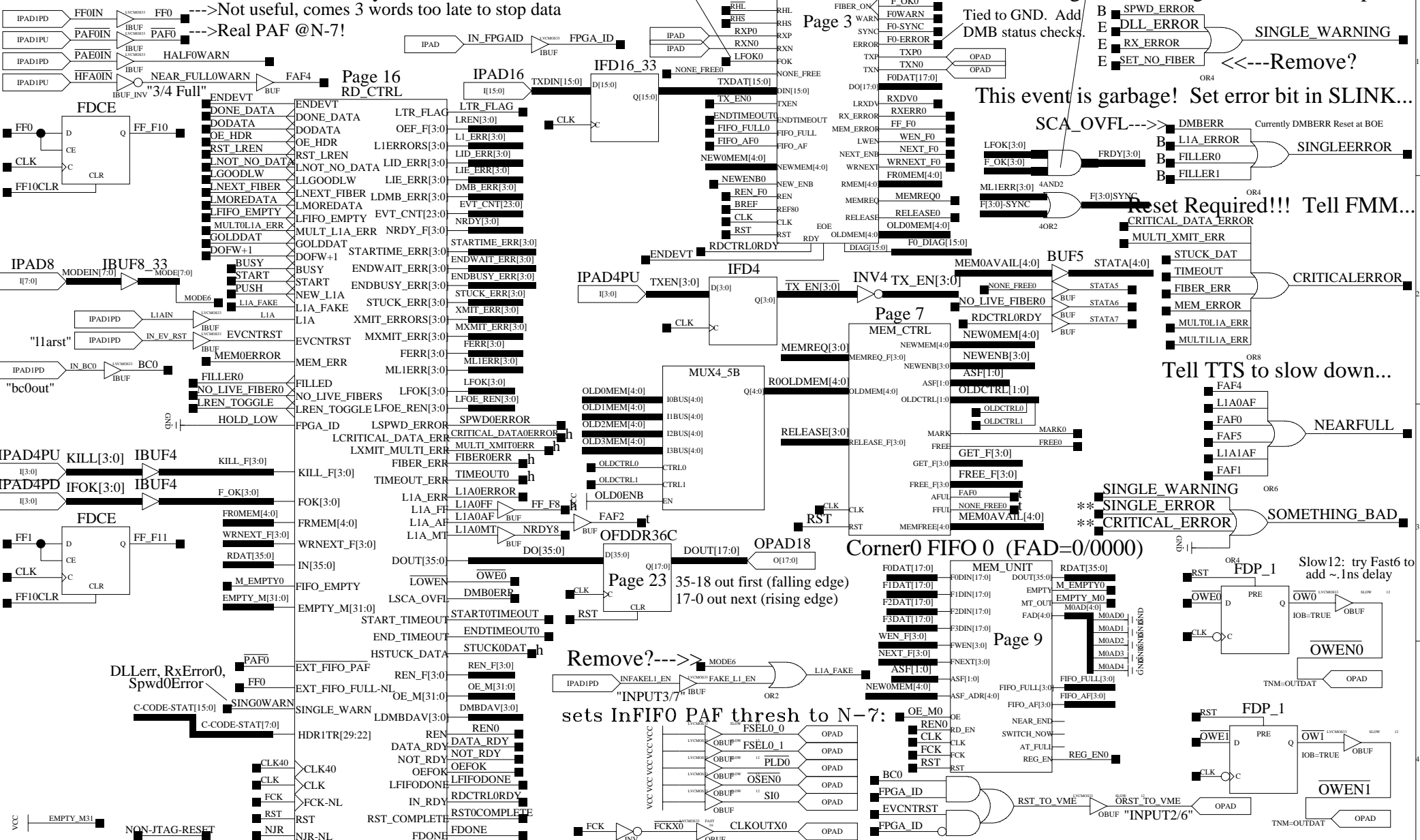
DDU WC, 11 DMB with 1 CFEB (nCFEB=11): 8CAh = 2250 dec, 18000 Bytes  
 DDU WC, 12 DMB with 1 CFEB (nCFEB=12): 996h = 2454 dec, 19632 Bytes  
 DDU WC, 15 DMB with 1 CFEB (nCFEB=15): BFAh = 3066 dec, 24528 Bytes

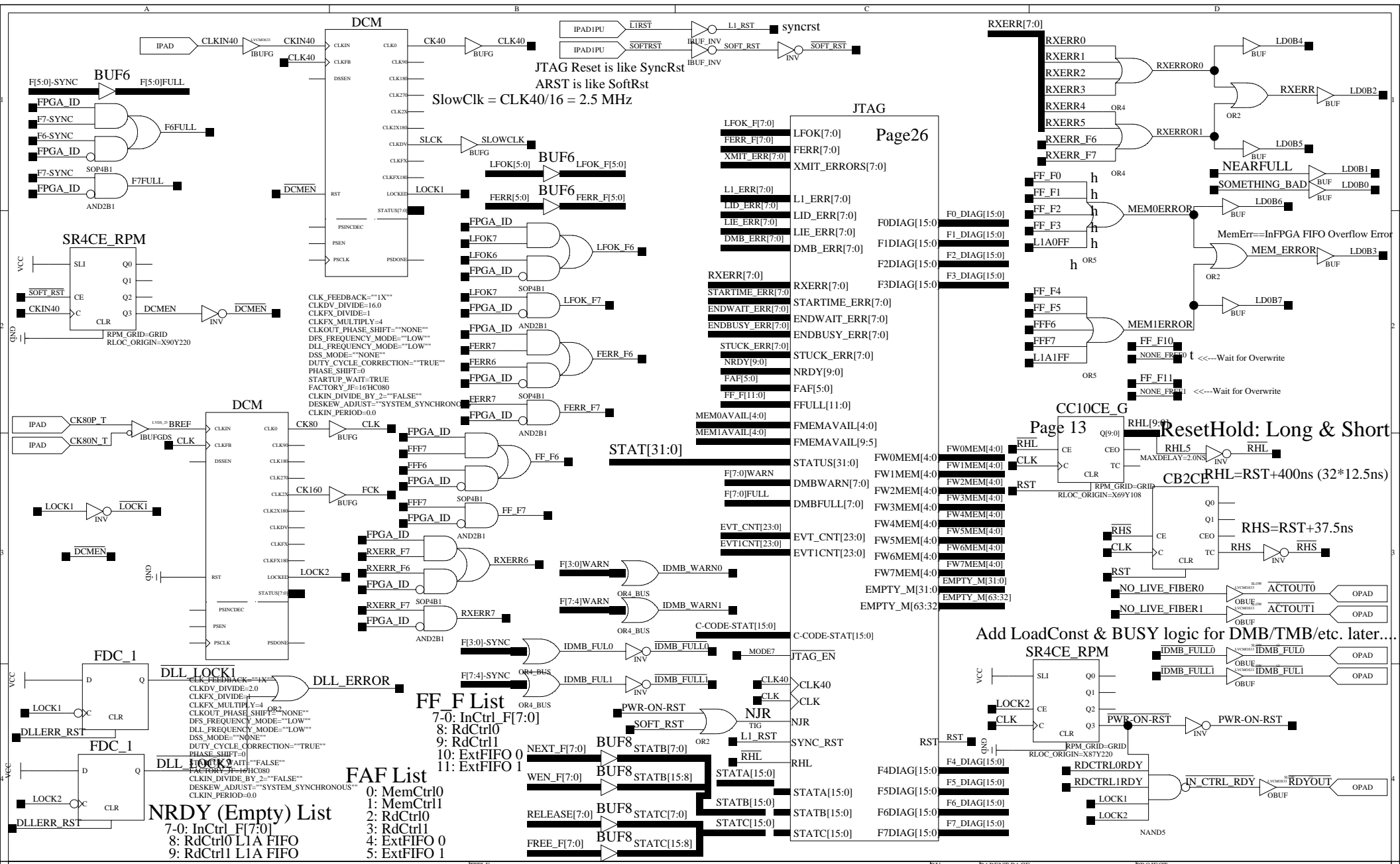
Later use FRDY logic to determine when InUnit is killed; allow real-time Kills (until Reset) in case of Errors/Glitches?

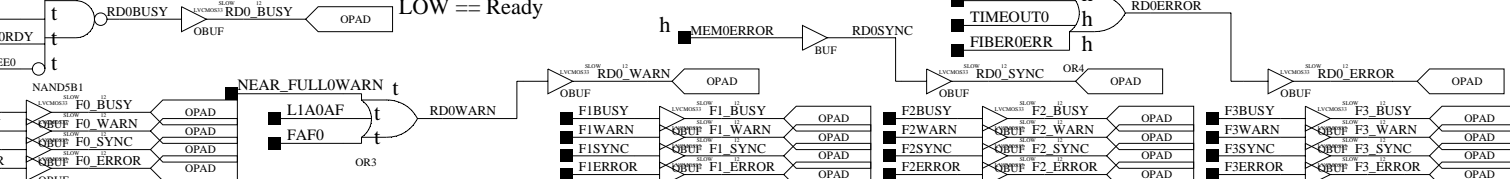
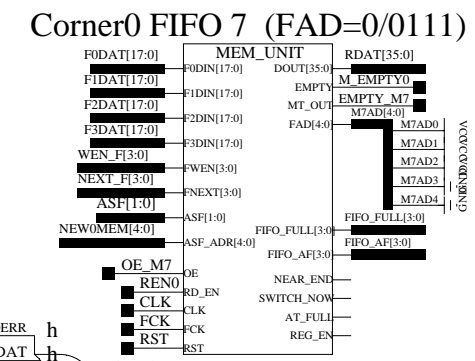
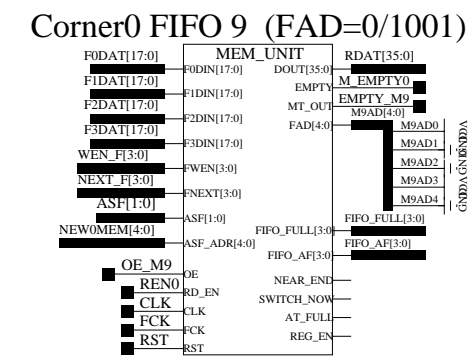
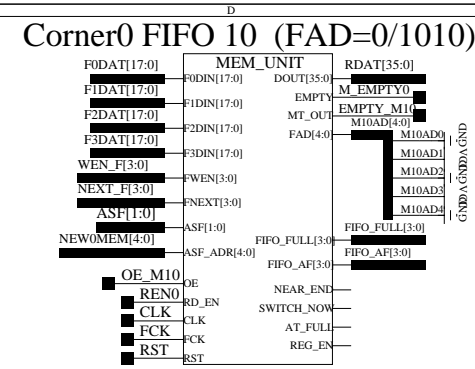
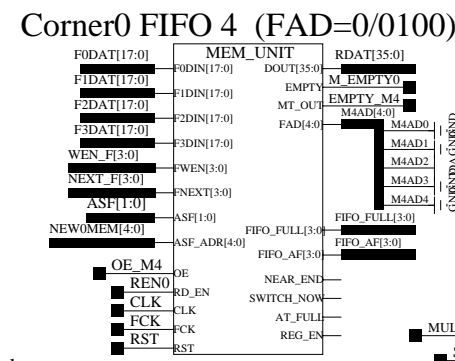
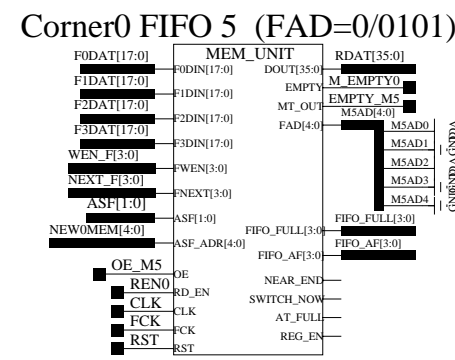
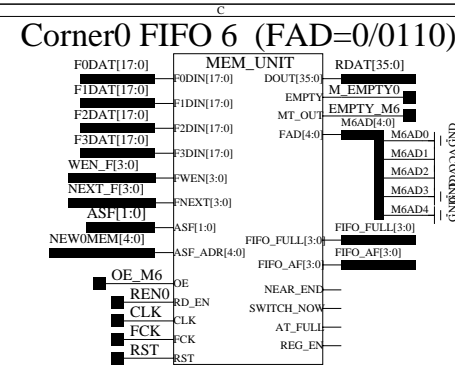
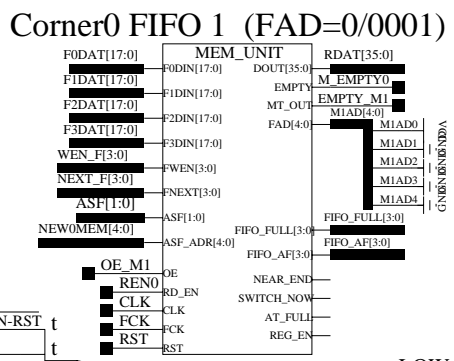
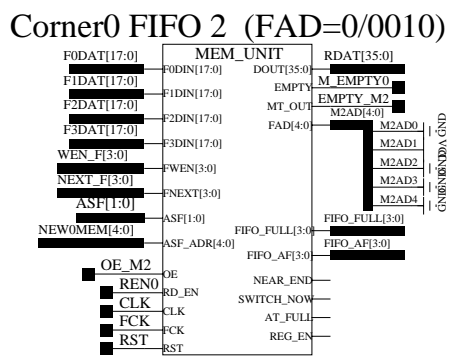
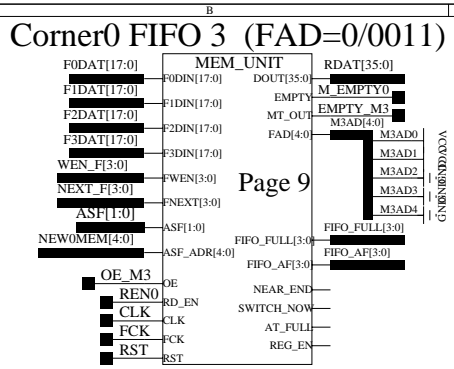
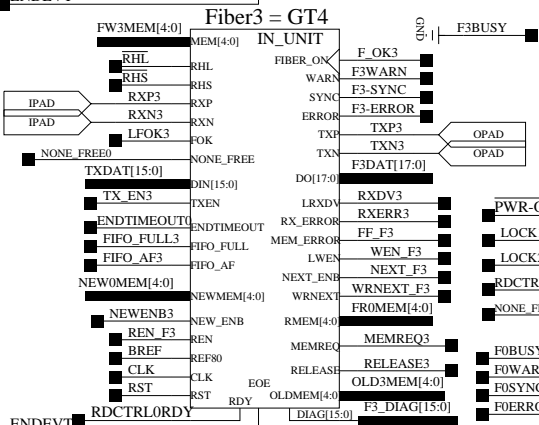
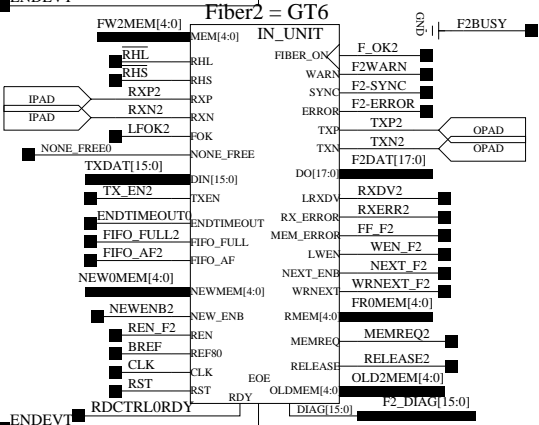
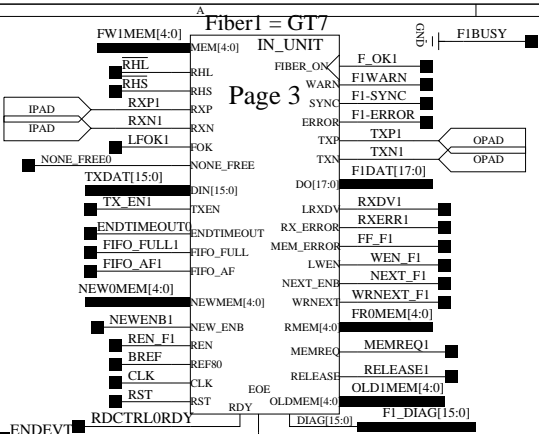
Mask DMBs with critical error until they're Reset?

For now just use LFOK.

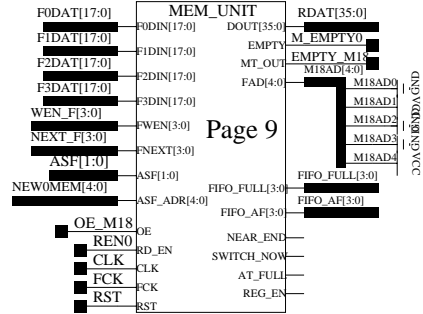
Single error, insignificant unless repeated



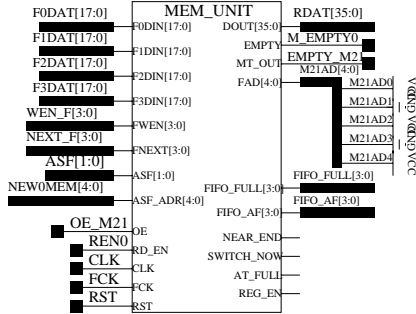




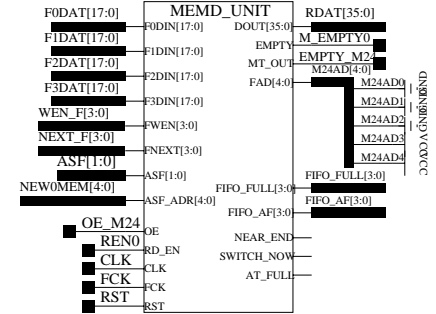
Corner1 FIFO 2 (FAD=1/0010)



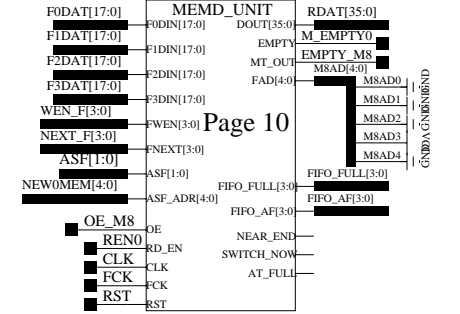
Corner1 FIFO 5 (FAD=1/0101)



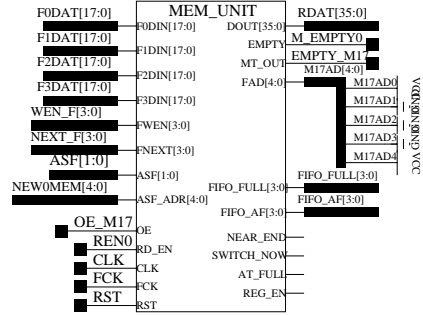
Corner1 FIFO 8\* (FAD=1/1000)



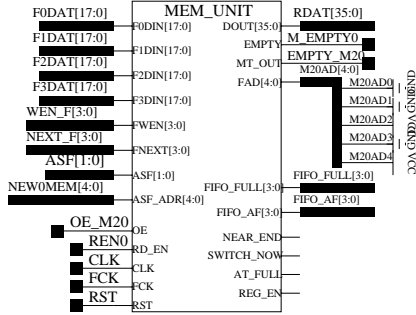
Corner0 FIFO 8\* (FAD=0/1000)



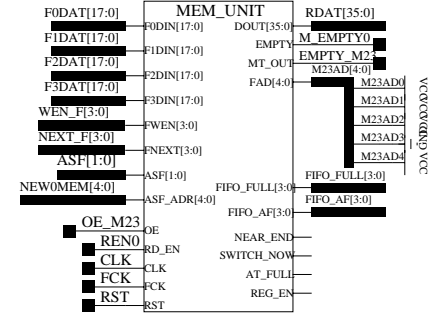
Corner1 FIFO 1 (FAD=1/0001)



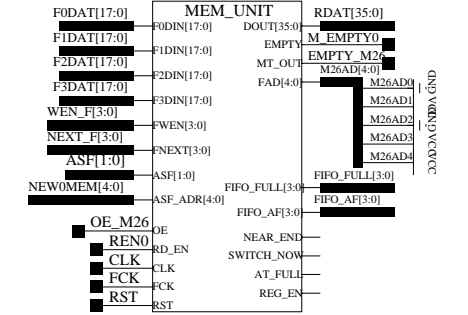
Corner1 FIFO 4 (FAD=1/0100)



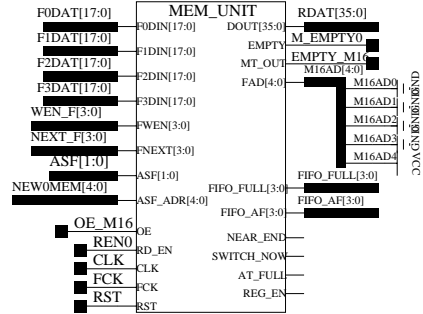
Corner1 FIFO 7 (FAD=1/0111)



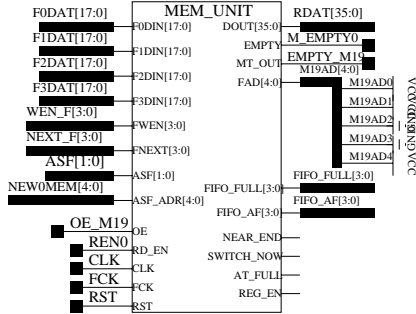
Corner1 FIFO 10 (FAD=1/1010)



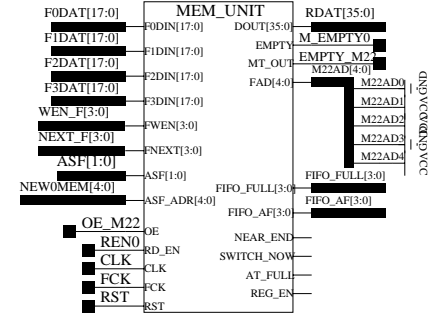
Corner1 FIFO 0 (FAD=1/0000)



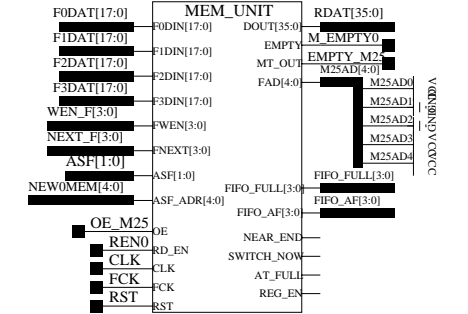
Corner1 FIFO 3 (FAD=1/0011)

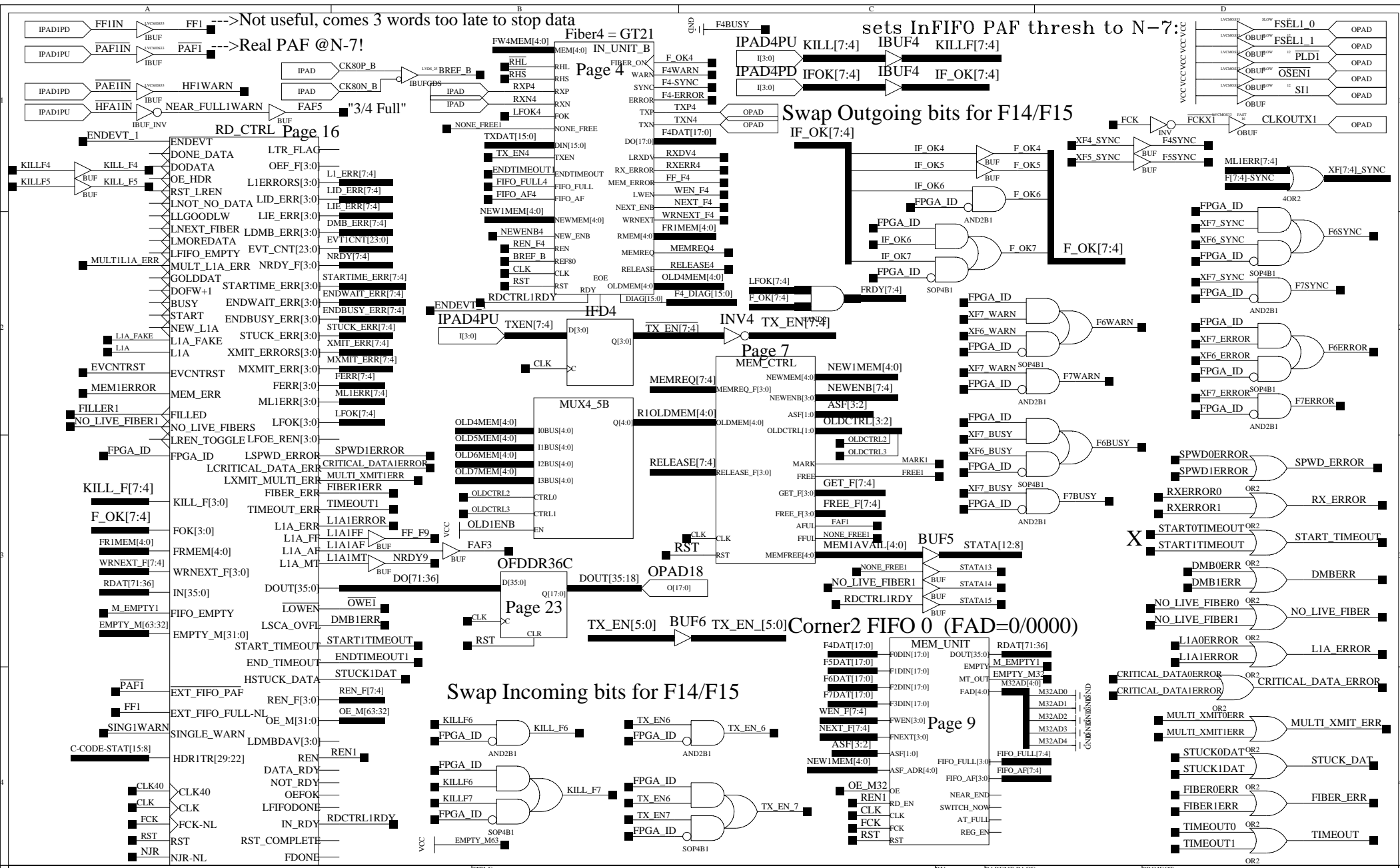


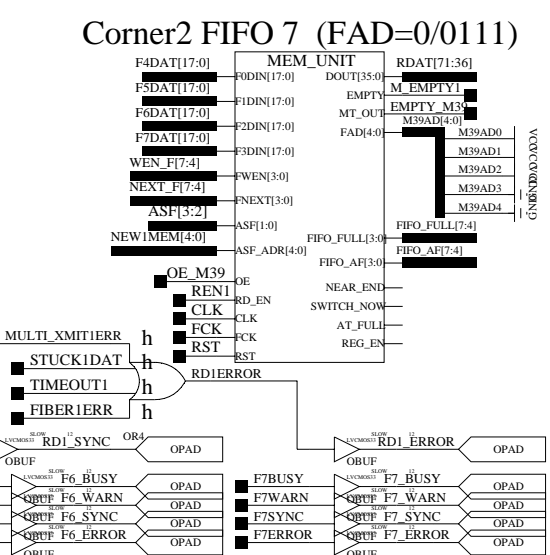
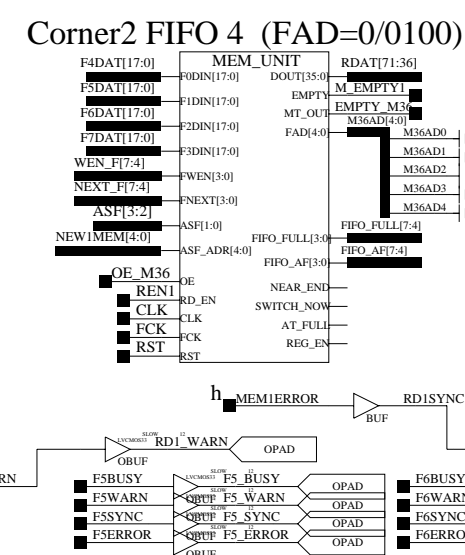
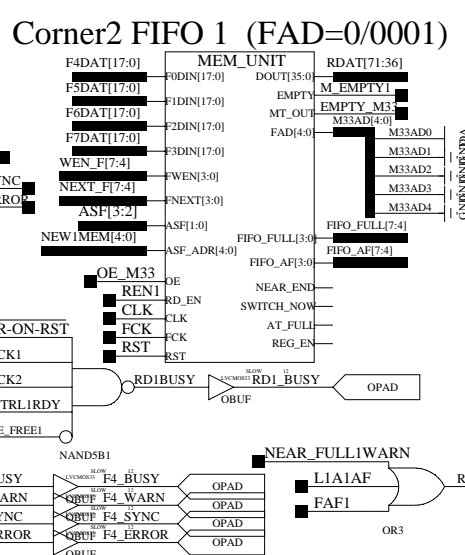
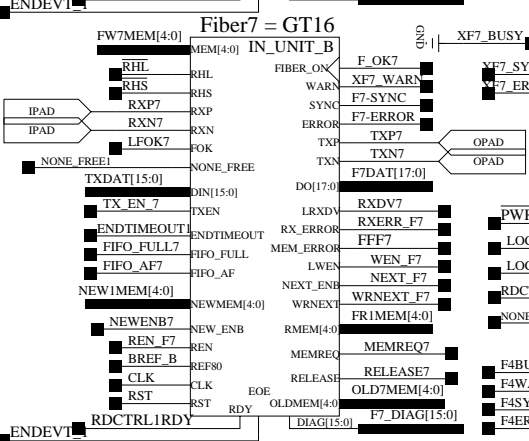
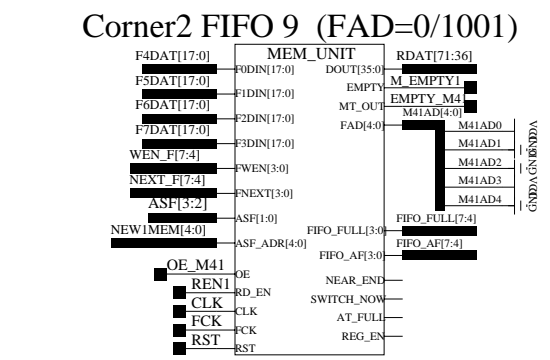
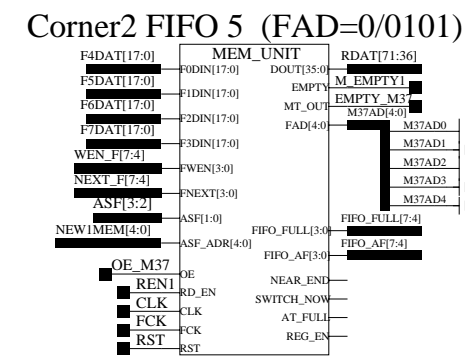
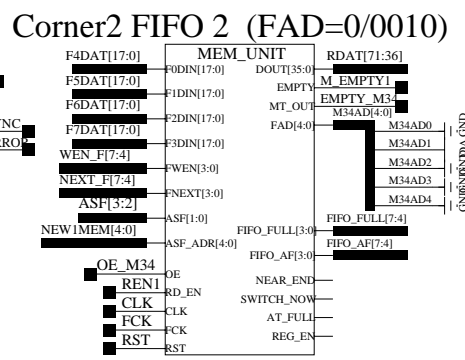
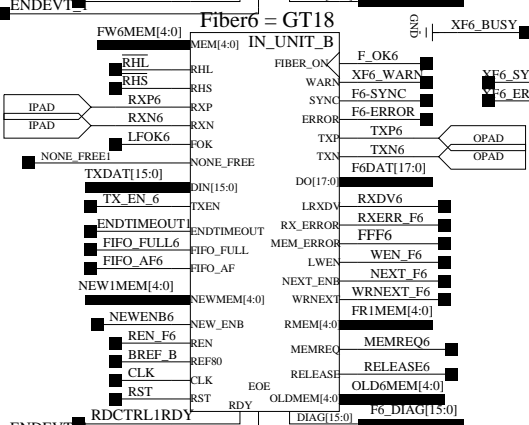
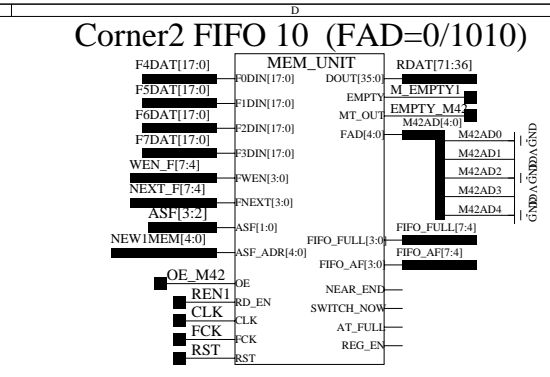
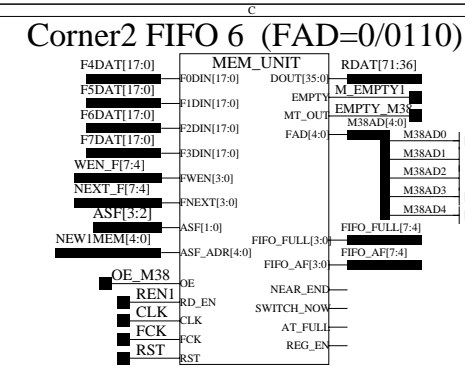
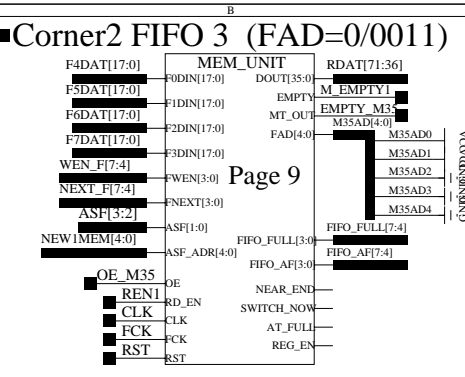
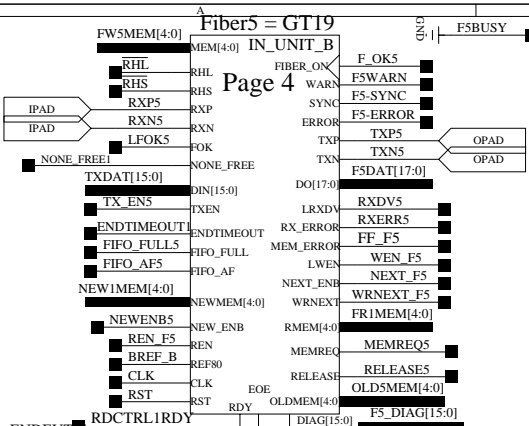
Corner1 FIFO 6 (FAD=1/0110)



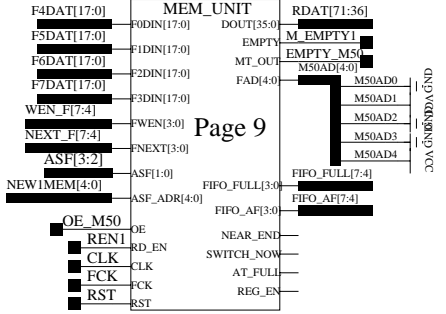
Corner1 FIFO 9 (FAD=1/1001)





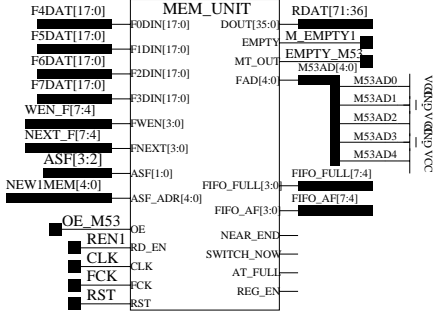


Corner3 FIFO 2 (FAD=1/0010)

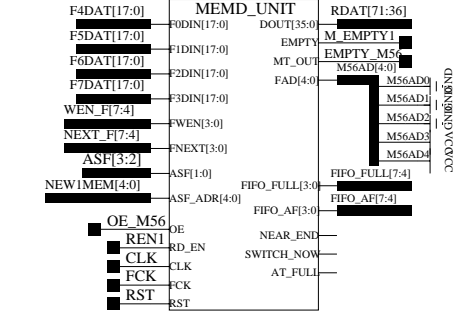


Page 9

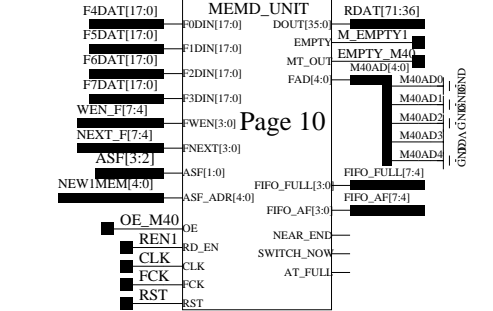
Corner3 FIFO 5 (FAD=1/0101)



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

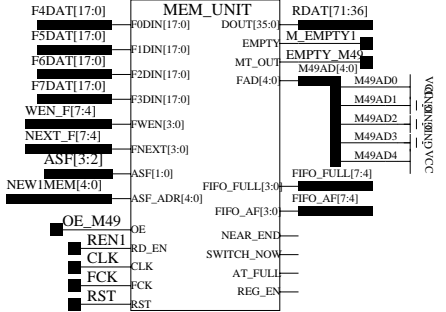


Corner2 FIFO 8\* (FAD=1/1000)

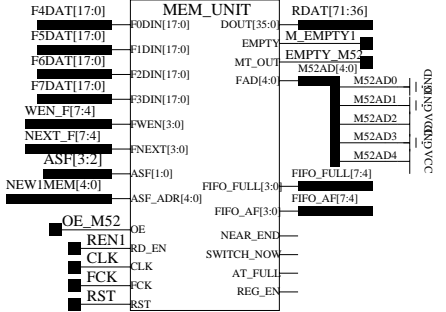


Page 10

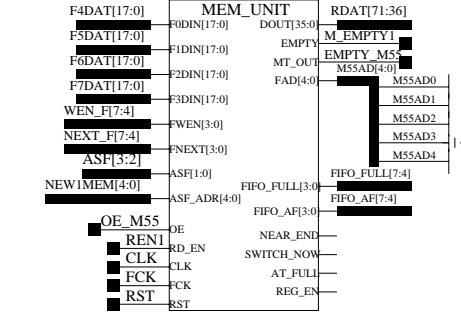
Corner3 FIFO 1 (FAD=1/0001)



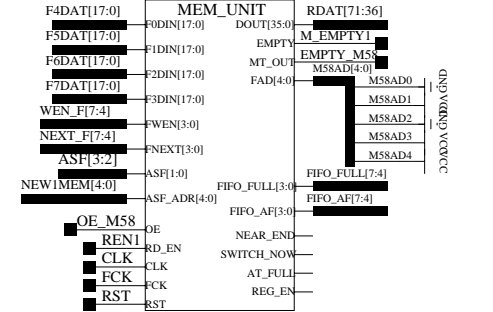
Corner3 FIFO 4 (FAD=1/0100)



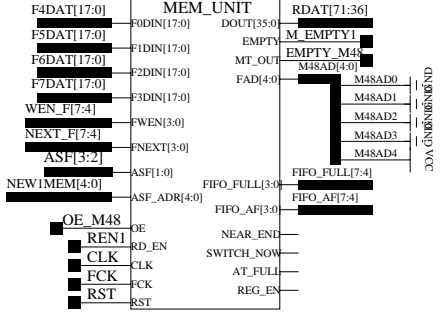
Corner3 FIFO 7 (FAD=1/0111)



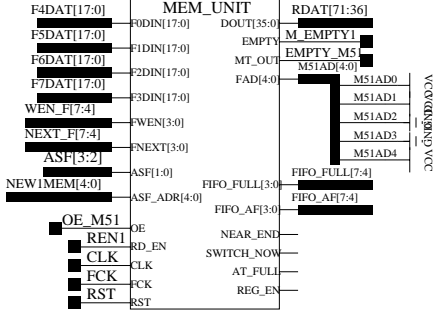
Corner3 FIFO 10 (FAD=1/1010)



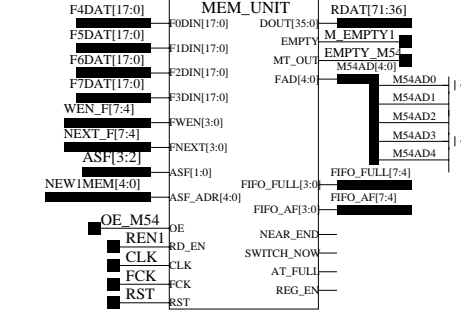
Corner3 FIFO 0 (FAD=1/0000)



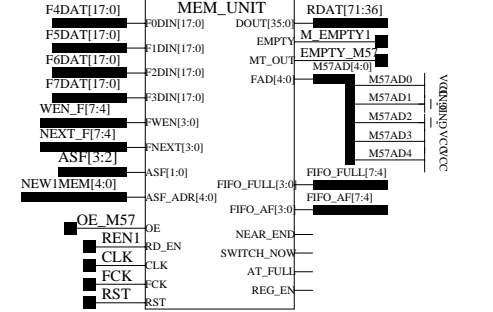
Corner3 FIFO 3 (FAD=1/0011)



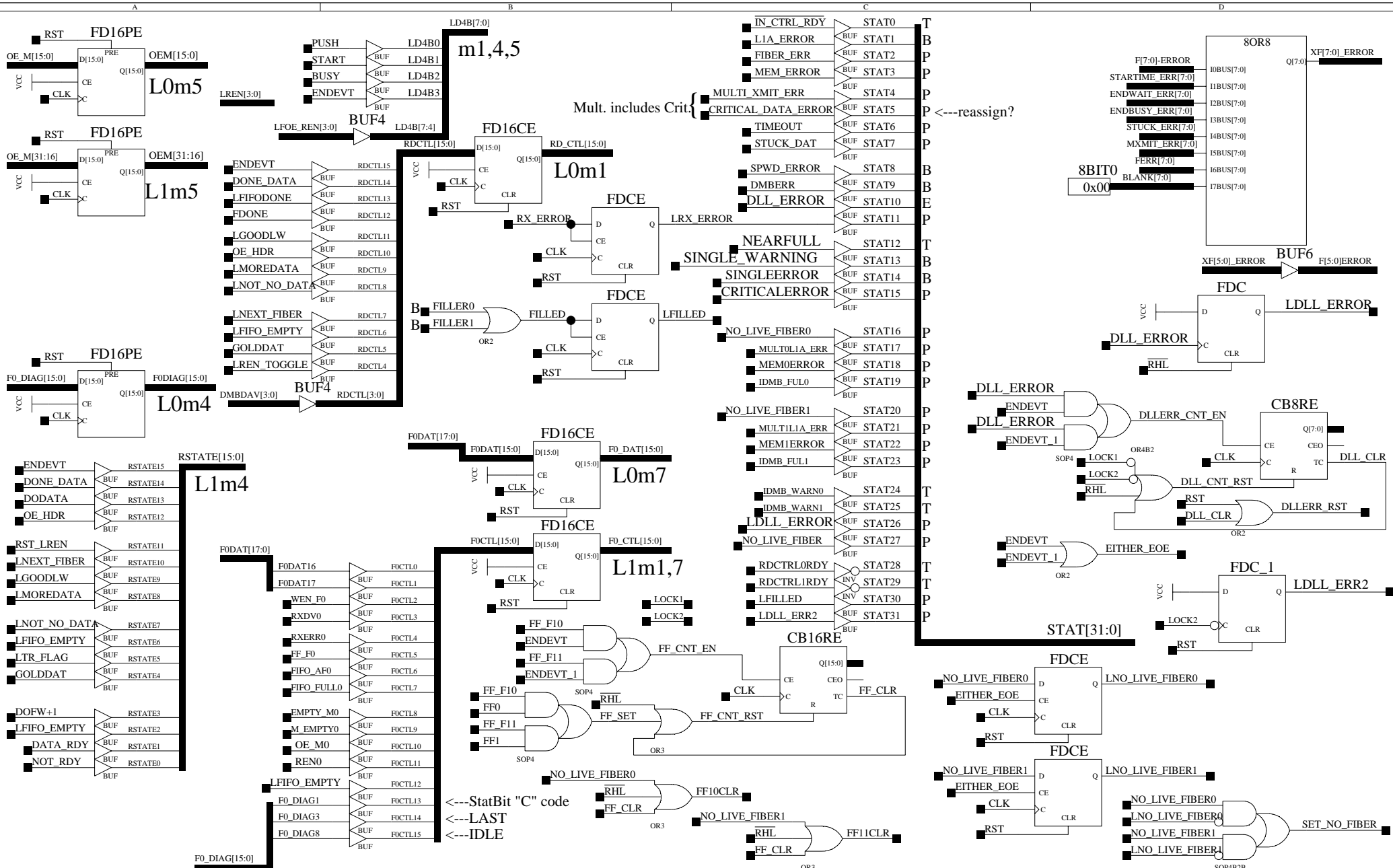
Corner3 FIFO 6 (FAD=1/0110)



Corner3 FIFO 9 (FAD=1/1001)



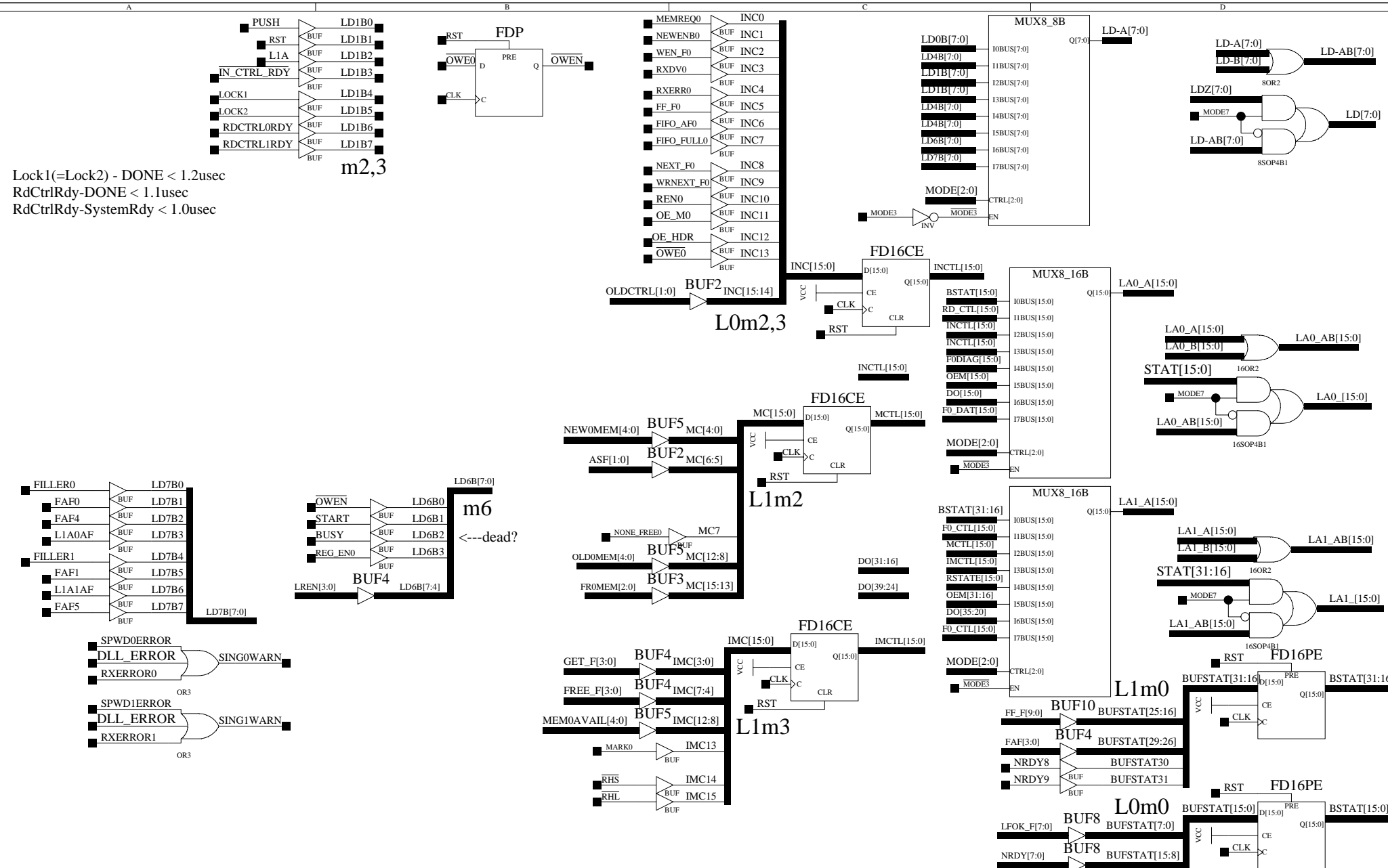




Mult. includes Crit

←←←reassign?

<---StatBit "C" code  
 <---LAST  
 <---IDLE



Lock1(=Lock2) - DONE < 1.2usec  
 RdCtrlRdy-DONE < 1.1usec  
 RdCtrlRdy-SystemRdy < 1.0usec

m2,3

L0m2,3

L1m2

L1m3

L1m0

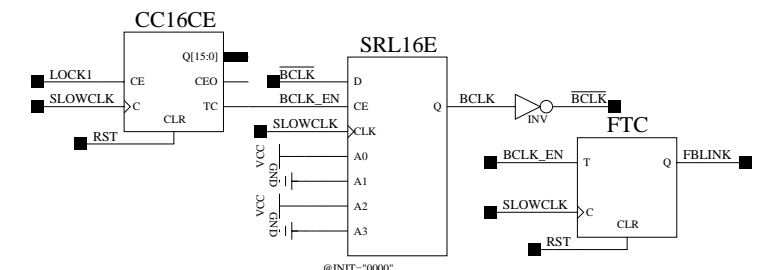
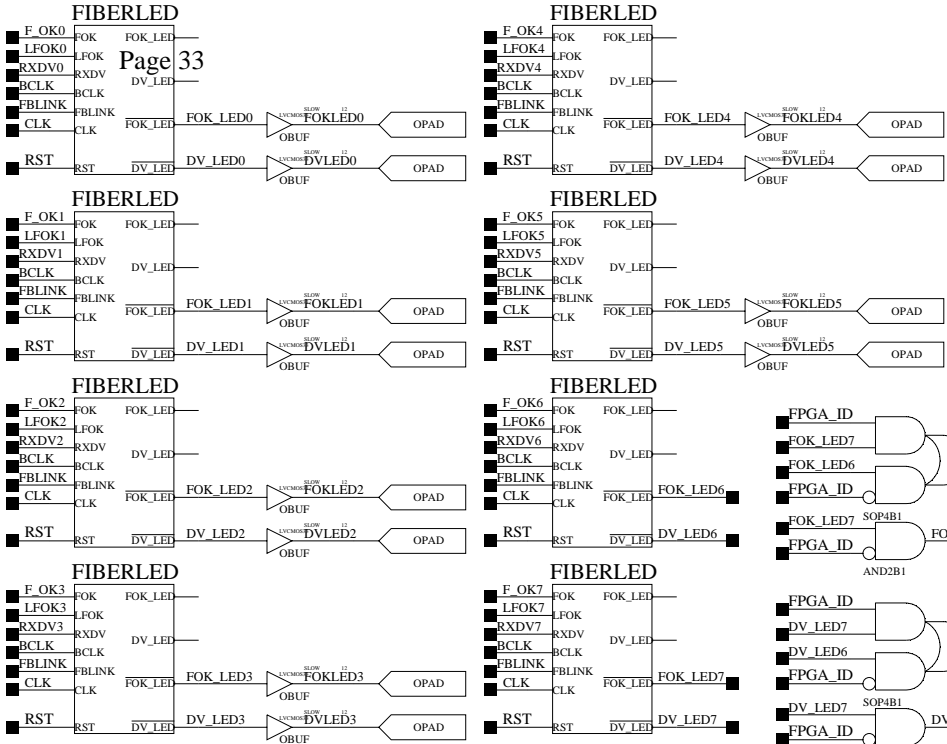
L0m0

m6

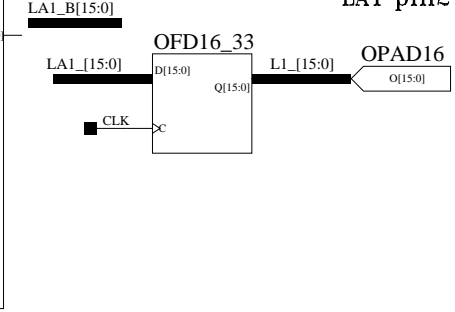
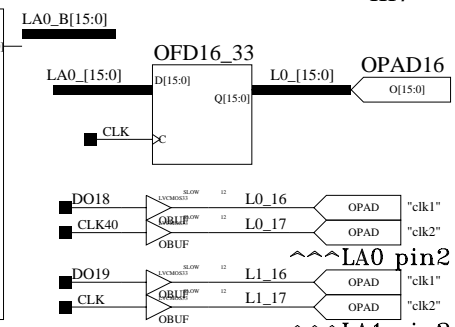
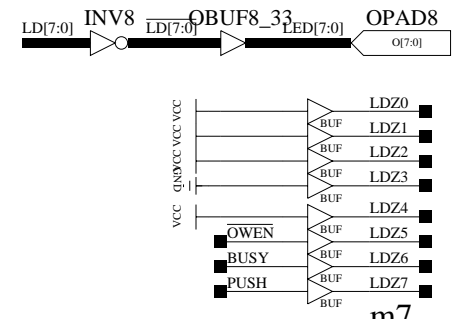
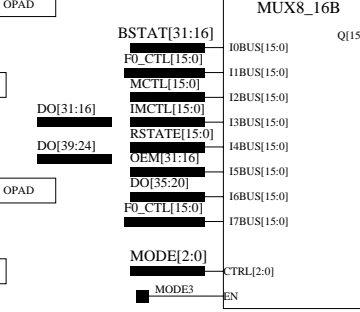
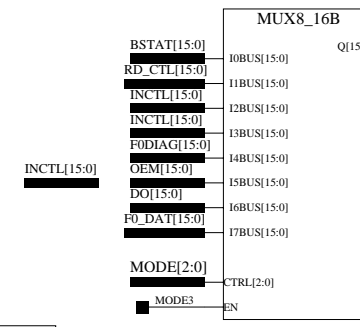
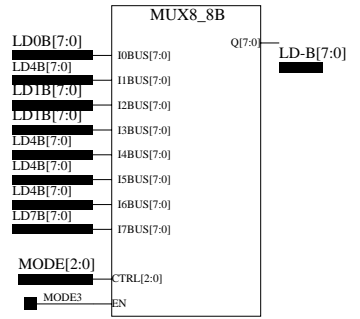
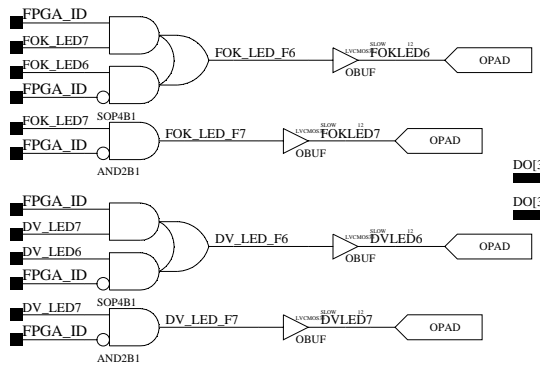
←---dead?

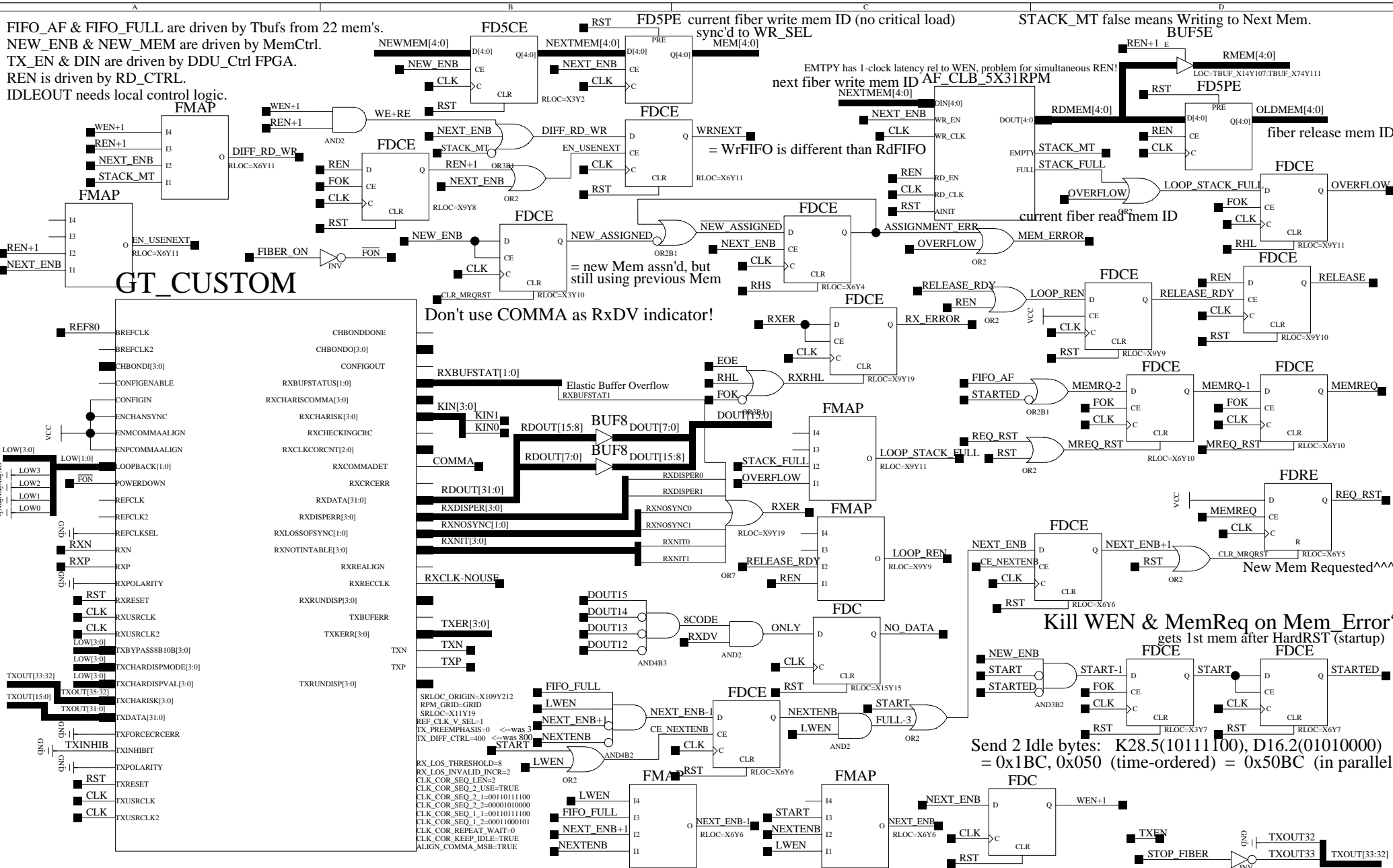
# END

Page 33

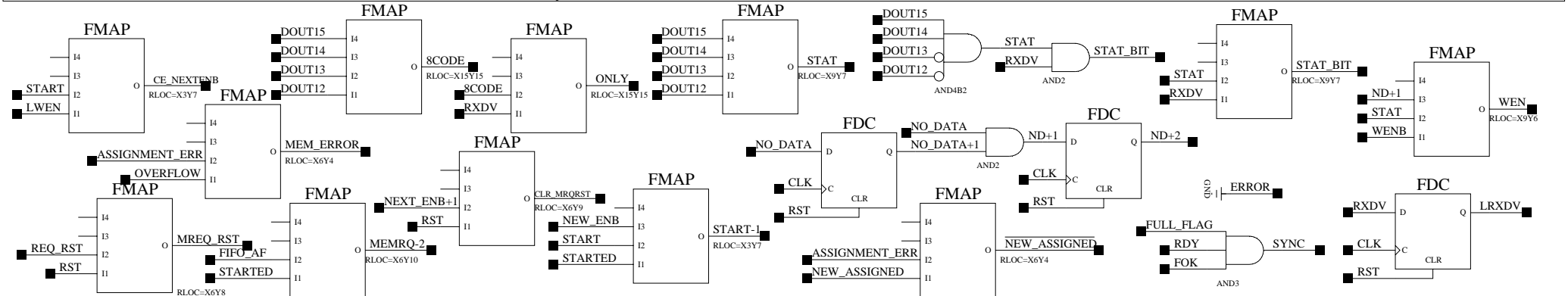
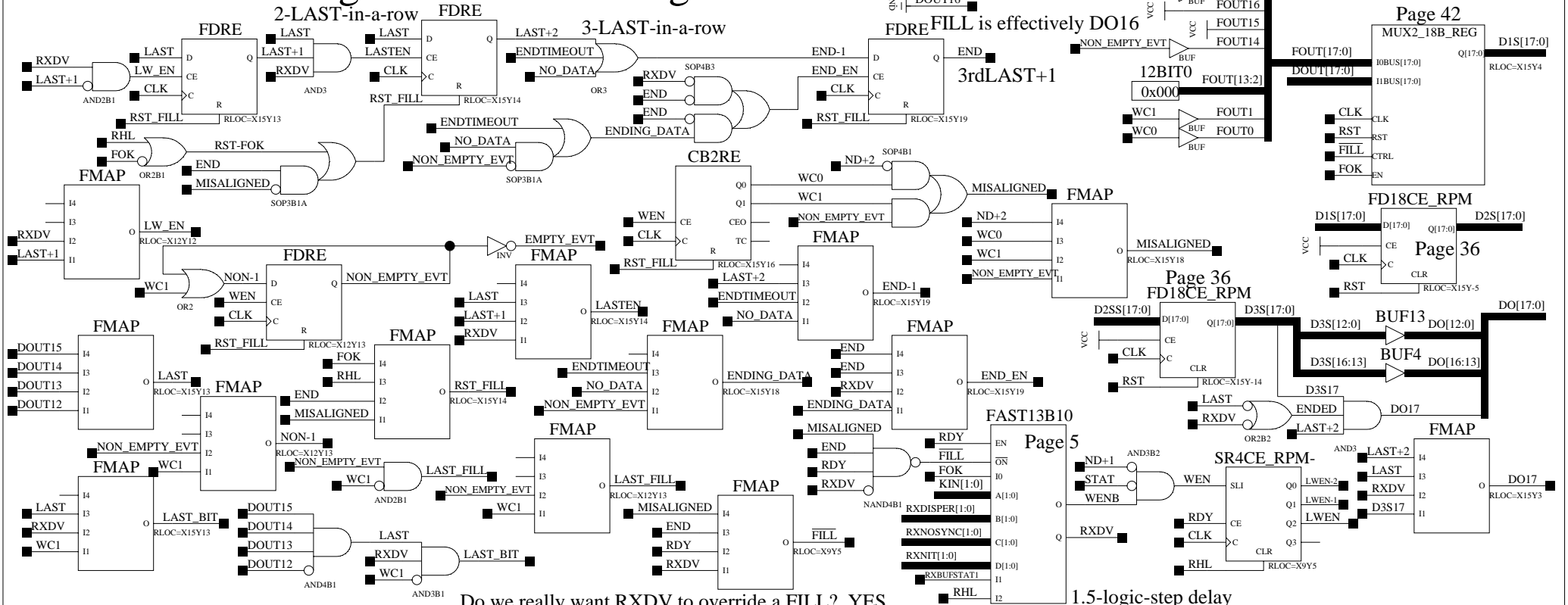


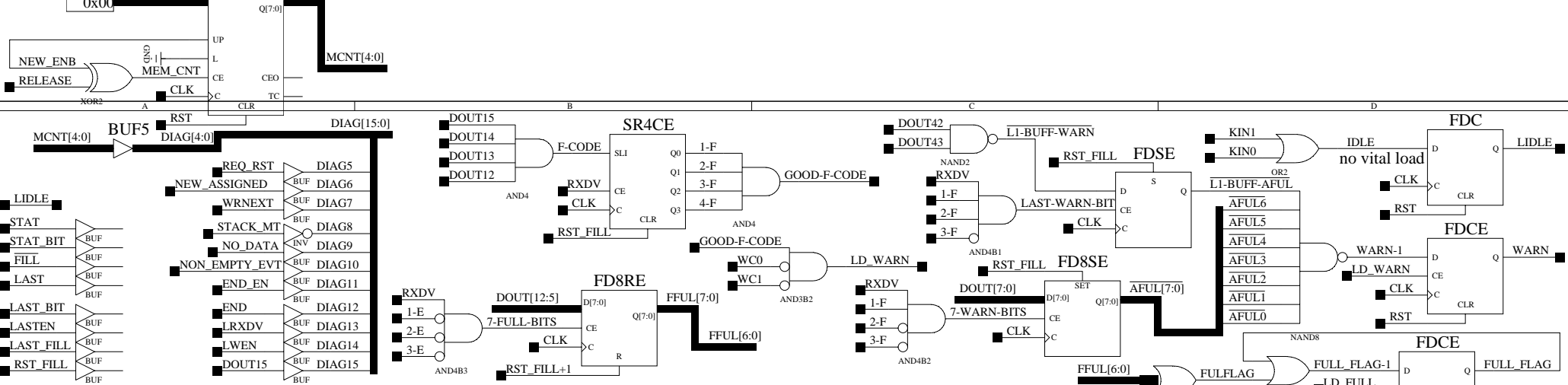
SlowCLK=2.5MHz, BCLK\_EN=38 Hz. Shift 13 will give ~1.5Hz BCLK, Shift 5 gives ~4Hz.





# FILL Logic for 2-4 word alignment





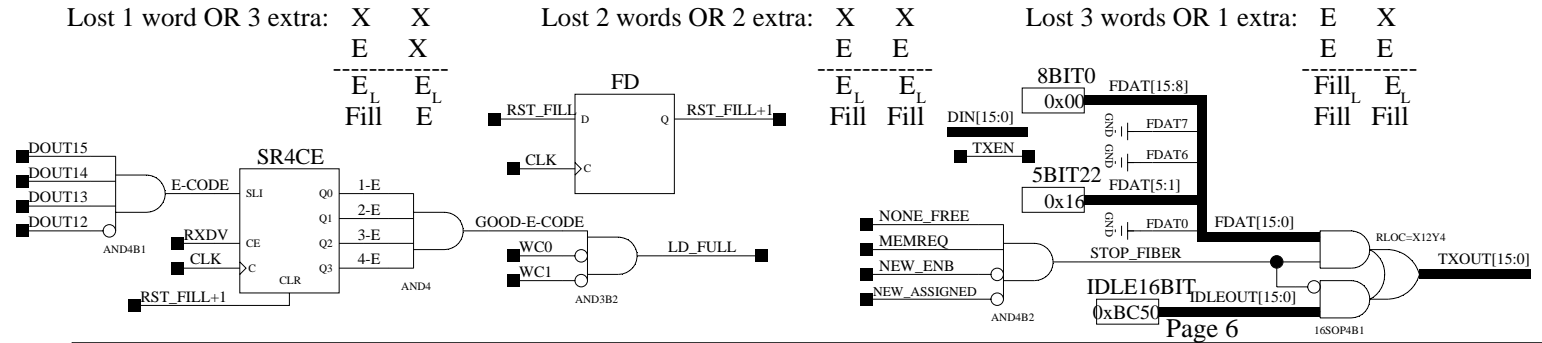
# LAST Flag is insensitive to single-bit errors

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

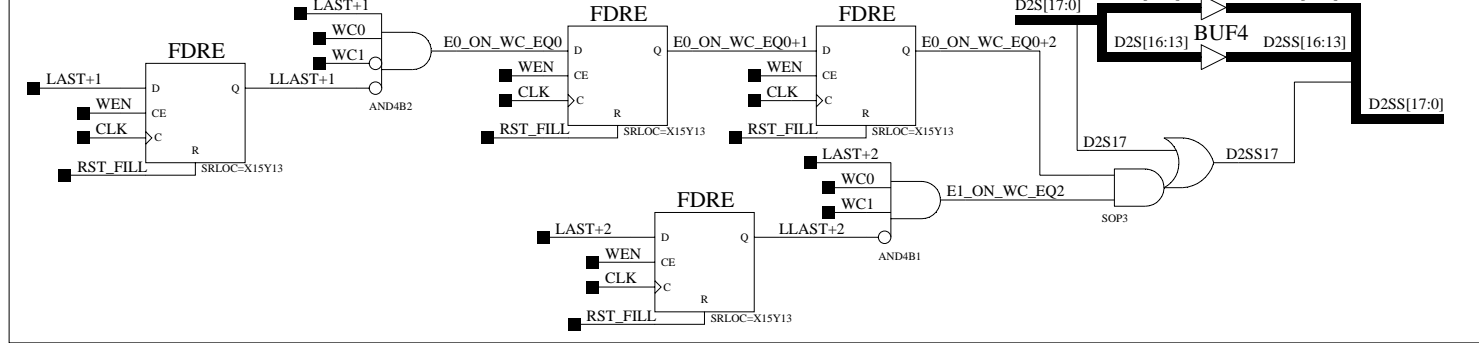
\*Assume 4 E-Codes at DMB End are sequential with no gaps\*

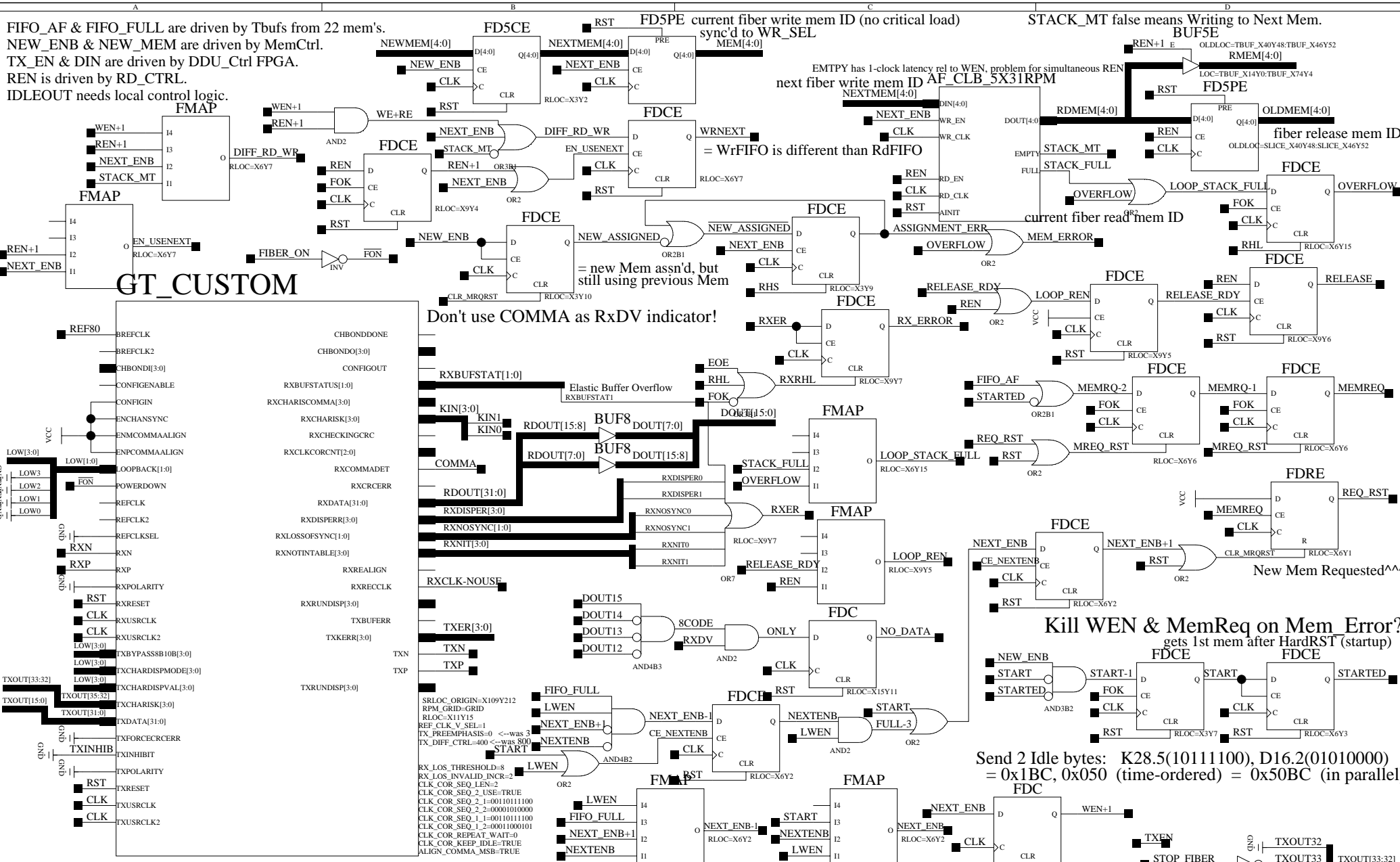
BRAM FIFO Output

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 <sub>L</sub>
	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		



### 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.

FD5PE current fiber write mem ID (no critical load) sync'd to WR\_SEL  
 MEM[4:0]

STACK\_MT false means Writing to Next Mem. BUF5E

EMPTPY has 1-clock latency rel to WEN, problem for simultaneous REN next fiber write mem ID AF\_CLB\_5X31RPM

WRNEXT = WrFIFO is different than RdFIFO

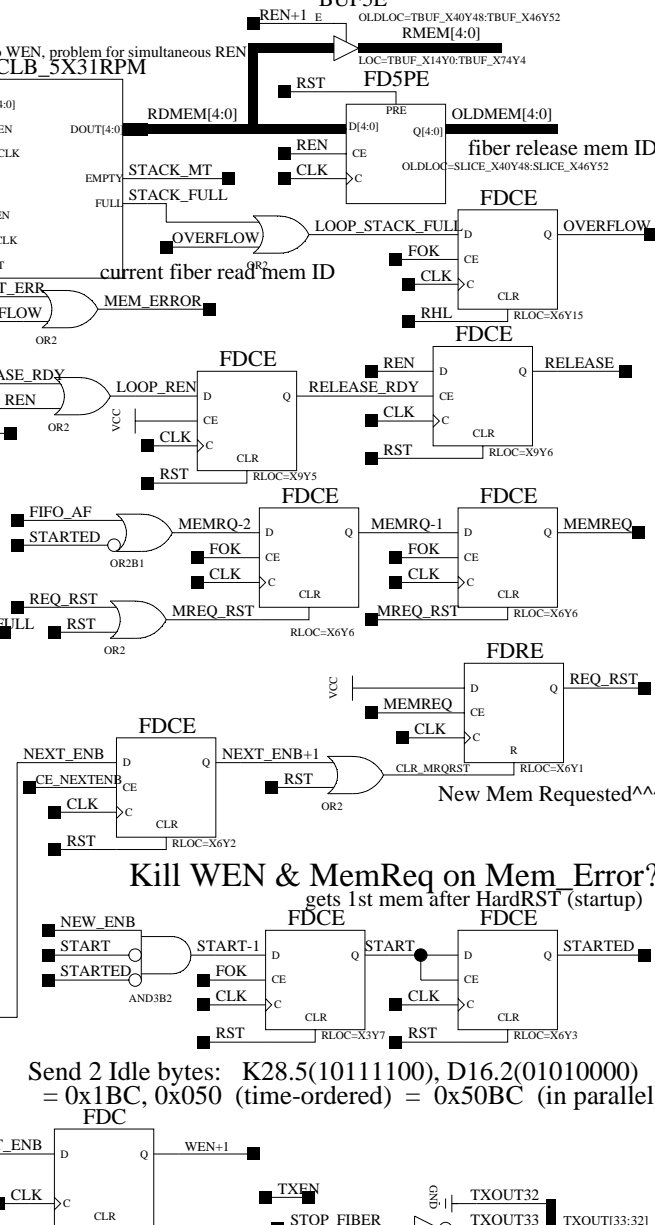
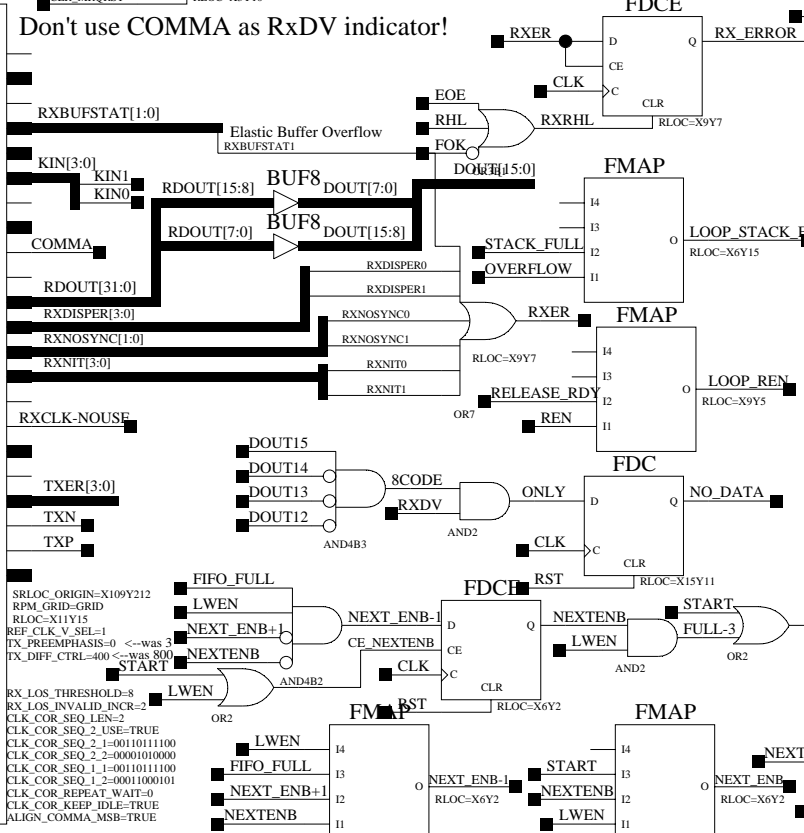
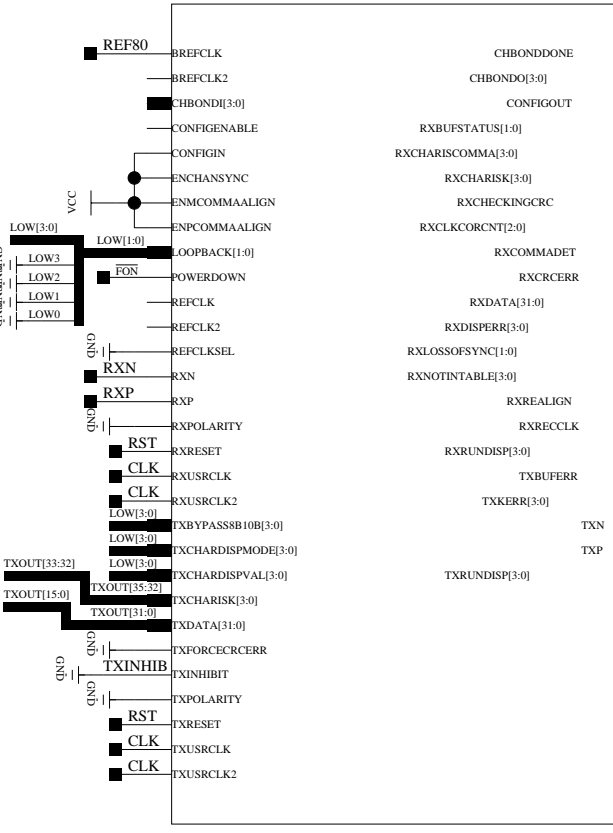
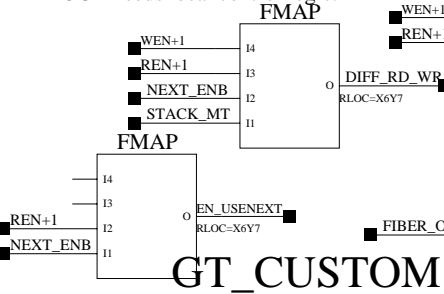
current fiber read mem ID

NEW\_ASSIGNED = new Mem assn'd, but still using previous Mem

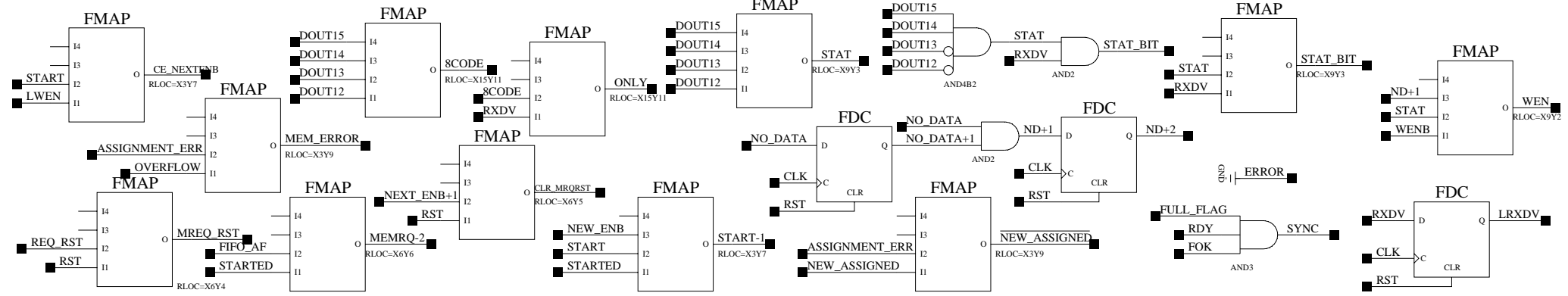
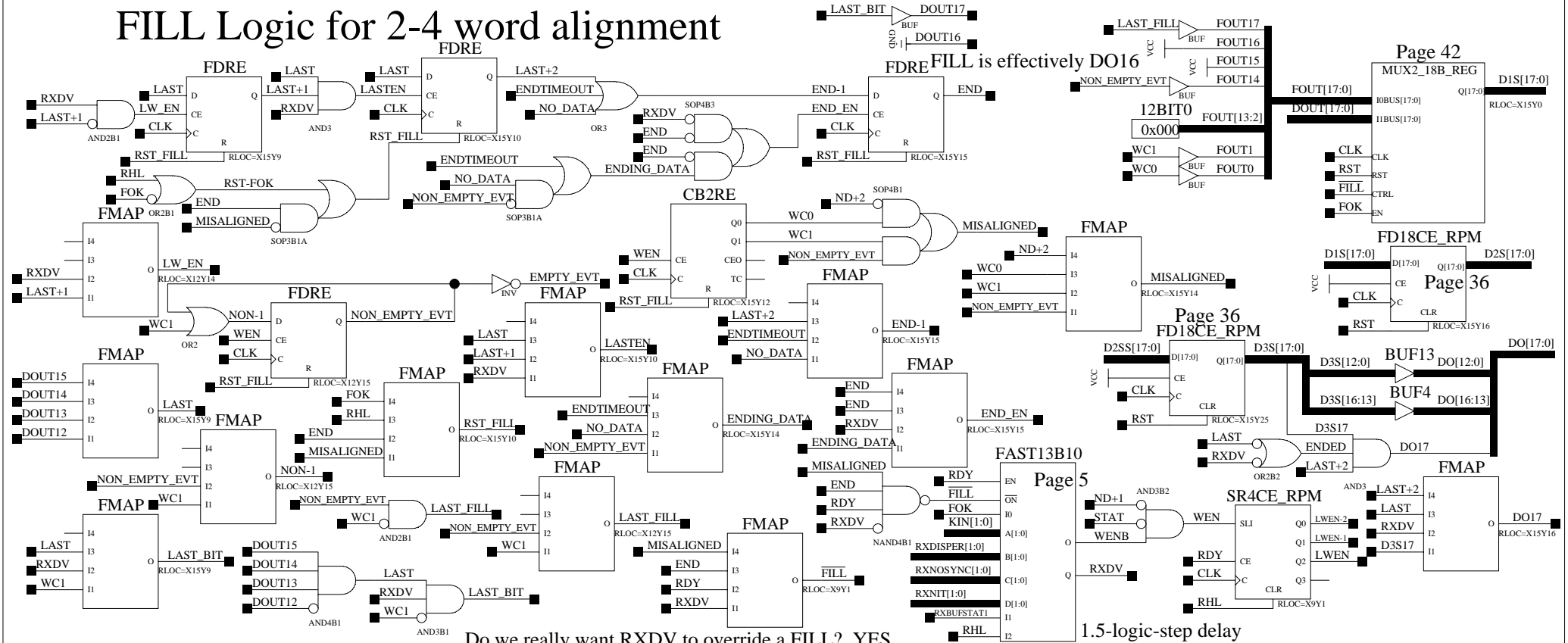
Don't use COMMA as RxDV indicator!

Kill WEN & MemReq on Mem\_Error? gets 1st mem after HardRST (startup)

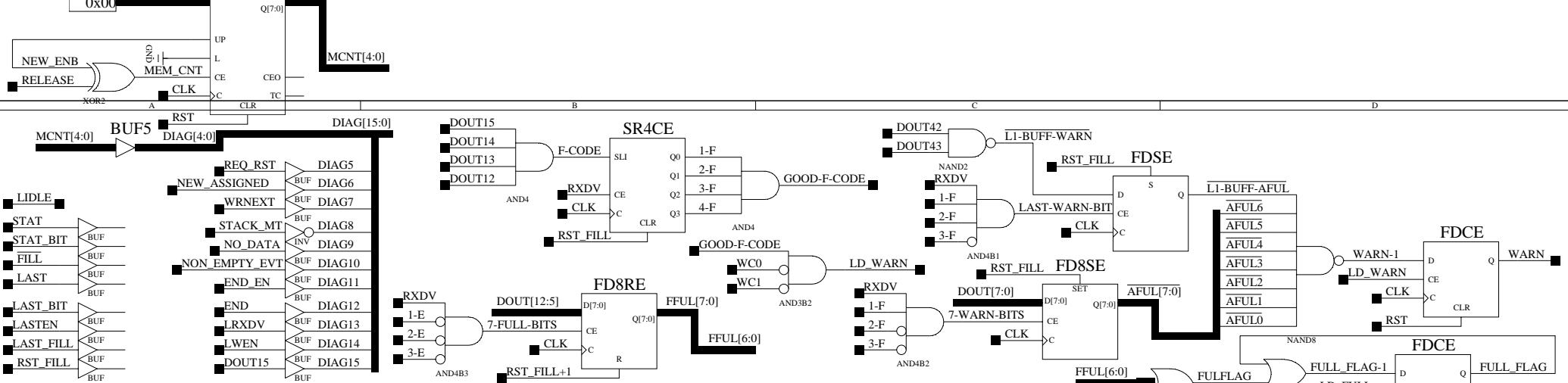
Send 2 Idle bytes: K28.5(10111100), D16.2(01010000) = 0x1BC, 0x050 (time-ordered) = 0x50BC (in parallel)



# 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
-----	
E <sub>L</sub>	E
E	E

Lost 1 word OR 3 extra:

X	X
E	X
-----	
E <sub>L</sub>	E <sub>L</sub>
Fill	E

Lost 2 words OR 2 extra:

X	X
E	E
-----	
E <sub>L</sub>	E <sub>L</sub>
Fill	Fill

Lost 3 words OR 1 extra:

E	X
E	E
-----	
Fill <sub>L</sub>	E <sub>L</sub>
Fill	Fill

Normal, but lose 1st E-Word  
(or lose last E-Word)

X	X
X	X
-----	
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
-----	
E <sub>L</sub>	E
Fill	E

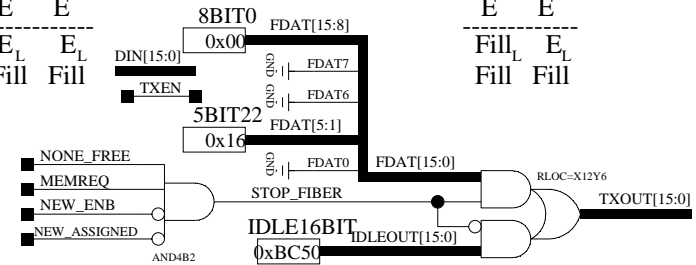
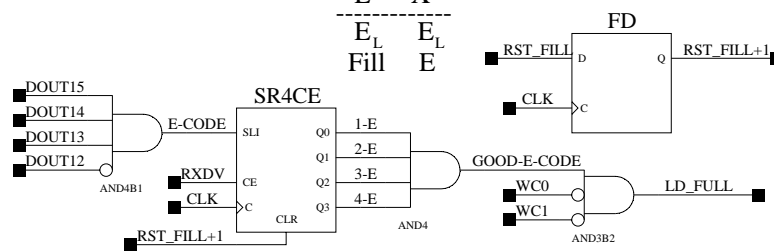
Normal, but BAD 1st E-Code:

X	X
X	X
-----	
E <sub>L</sub>	X
E	E

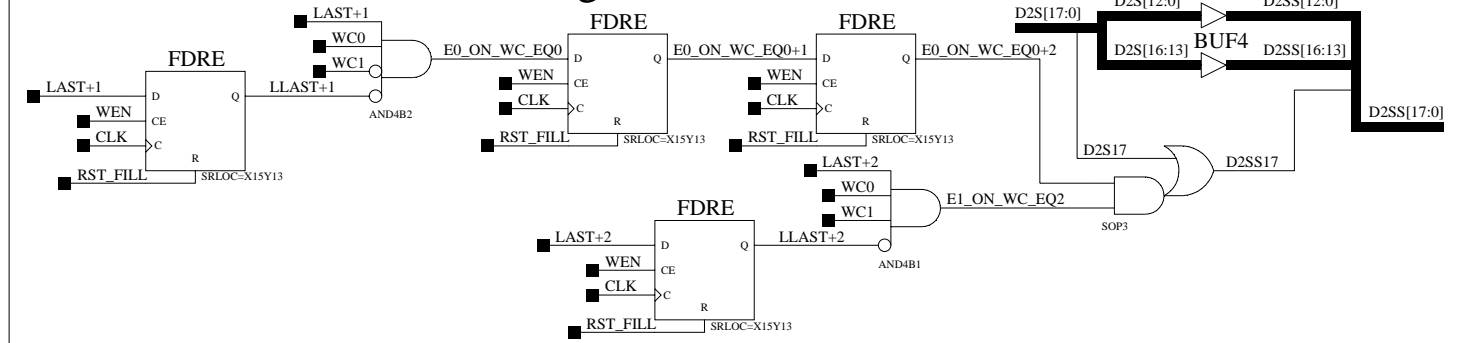
Normal, but BAD 2nd E-Code:

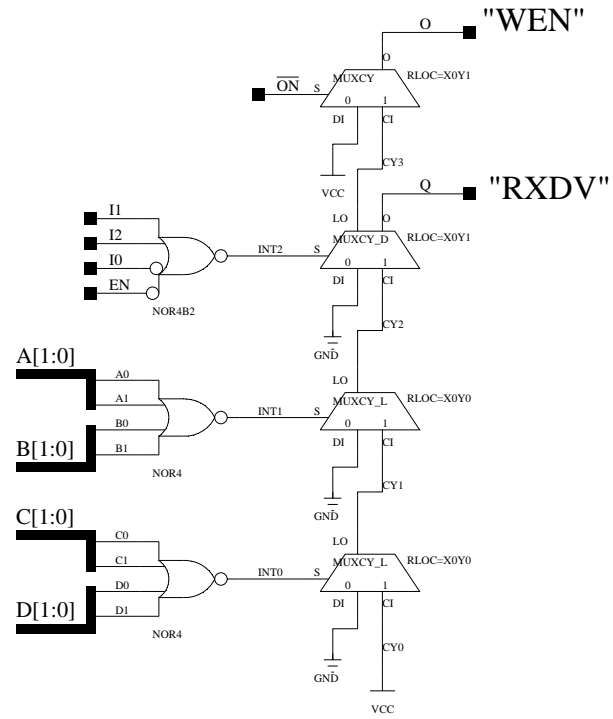
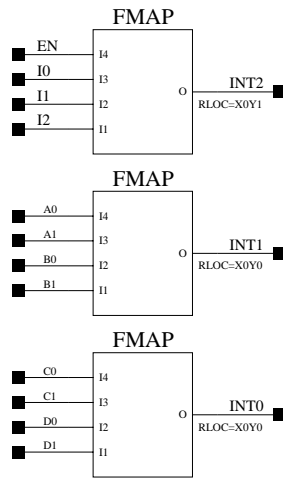
X	X
X	X
-----	
X <sub>L</sub>	E
E	E

BAD 3rd or 4th E-Code is no problem



Set LAST Flag for Bad 2nd E-Code case



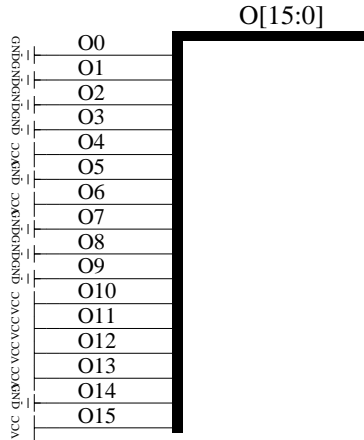


JRG

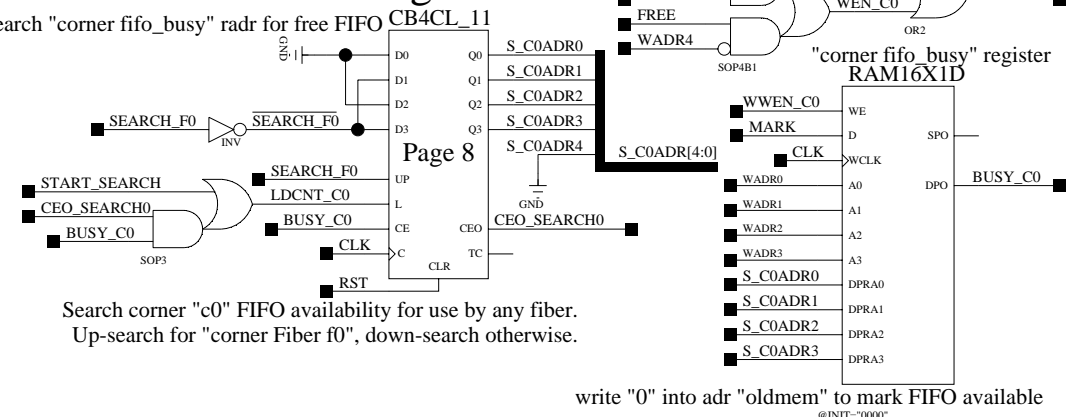
Title:	FAST13B10	
Comments:	Custom Logic for DDU similar to: AND12B10 with an OR2 (allows ON to override)	
Date:	19th December 2003	Ver: 1
Sheet Size:	B	Rev: A

Send 2 Idle bytes:

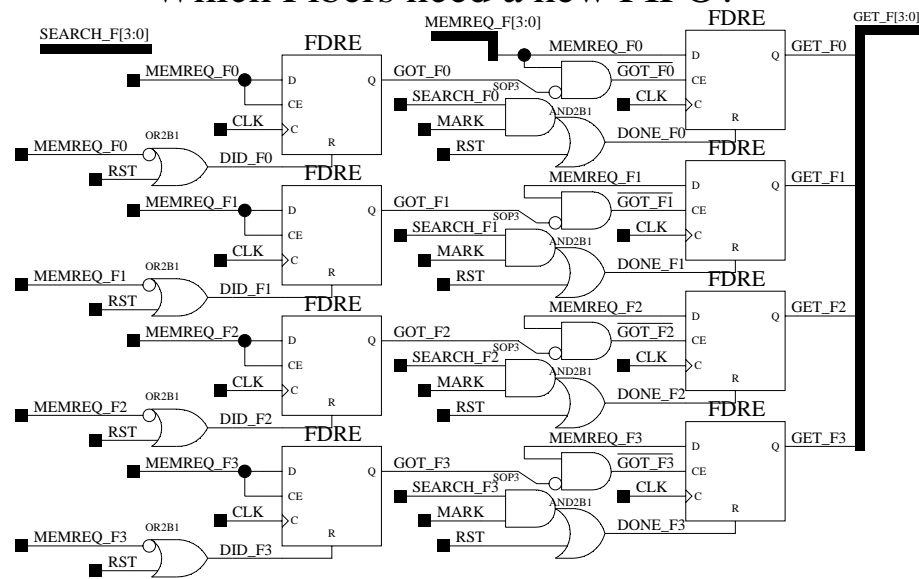
$K28.5(10111100)+D16.2(01010000)$   
= 0x1BC + 0x050 (time-ordered)  
= 0xBC50 (in parallel)



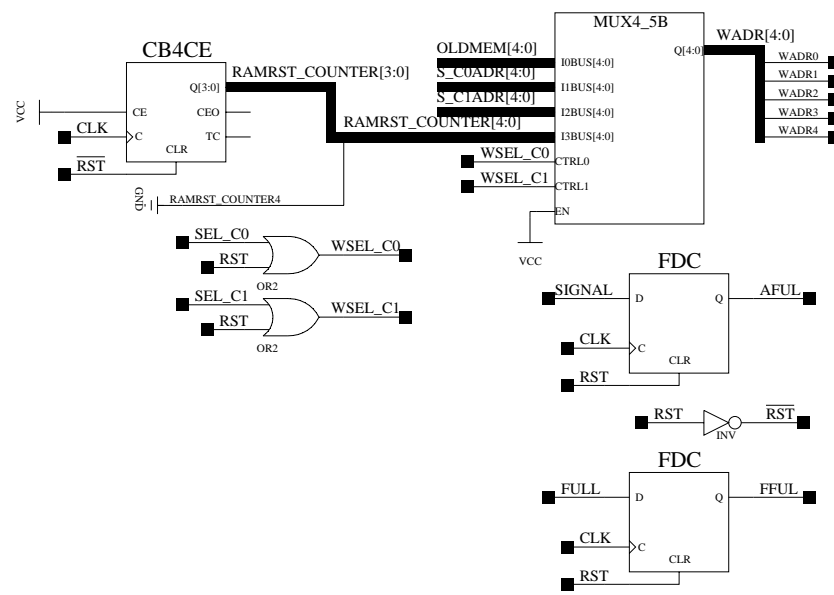
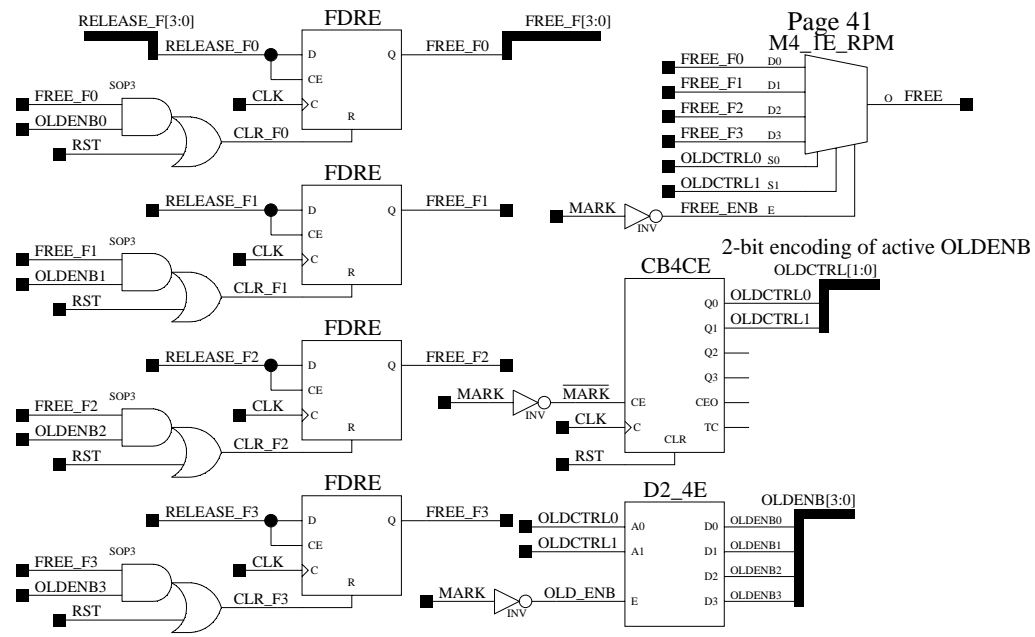
# Corner 0 FIFO usage



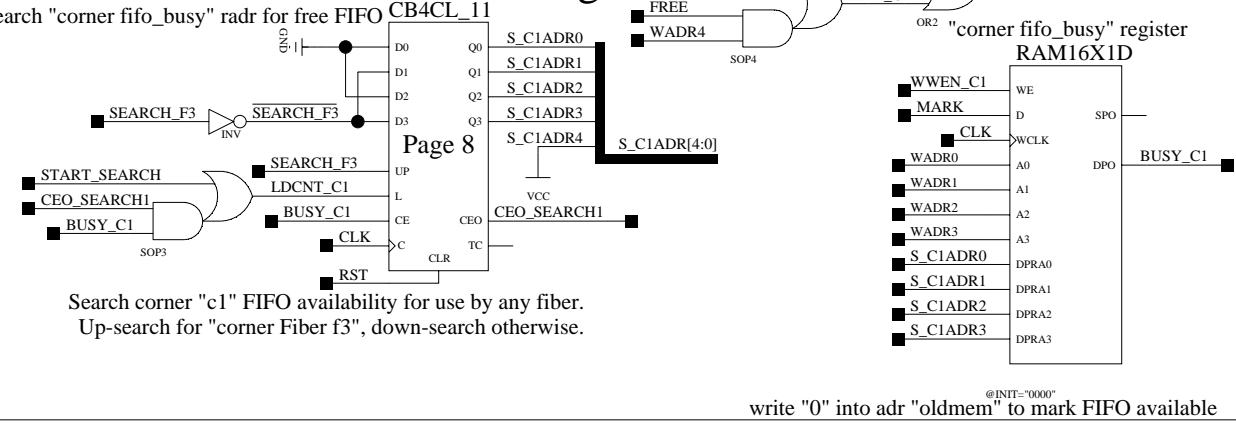
# Which Fibers need a new FIFO?



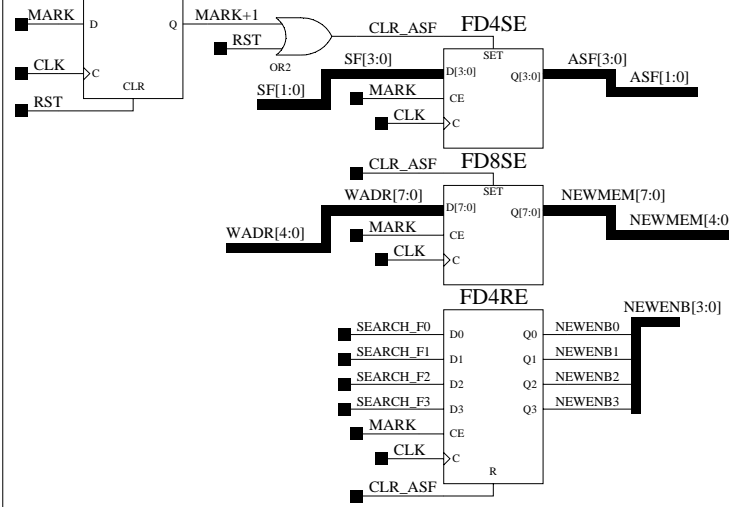
# FIFO Release Control



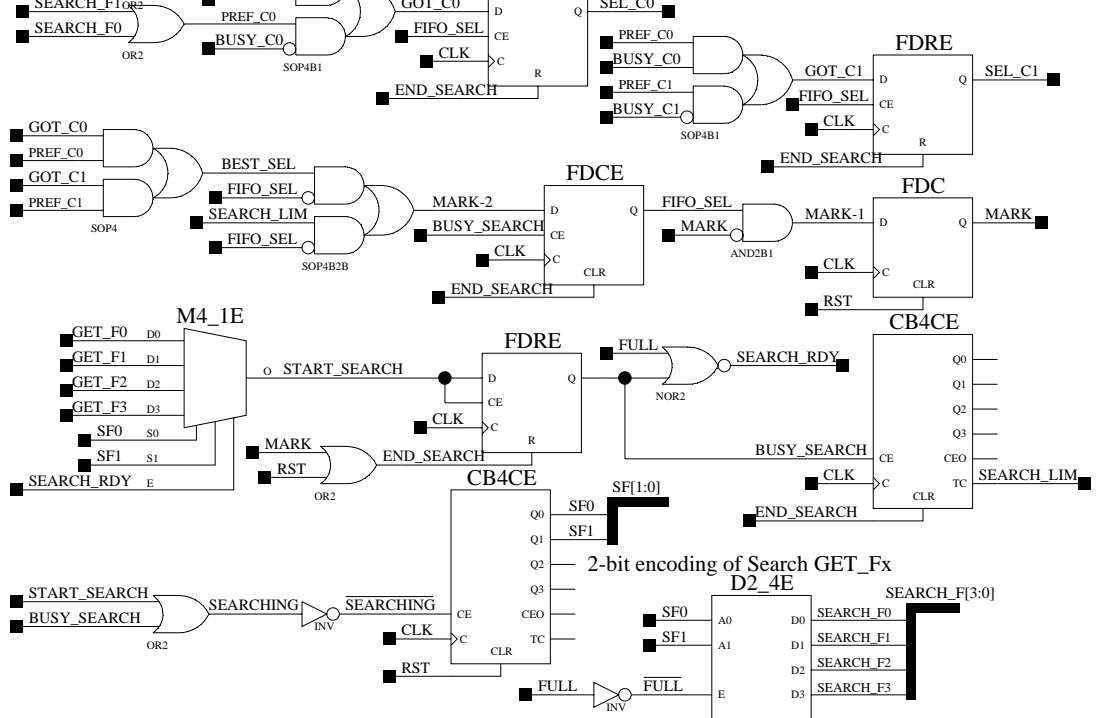
# Corner 1 FIFO usage



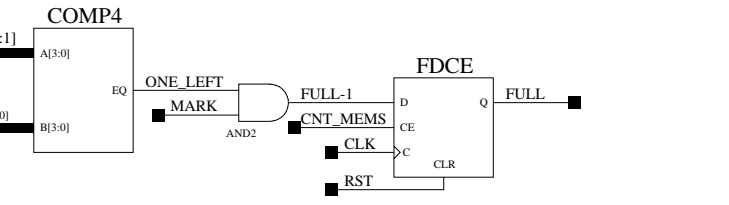
# Assign Fiber to FIFO & Assign FIFO to Fiber



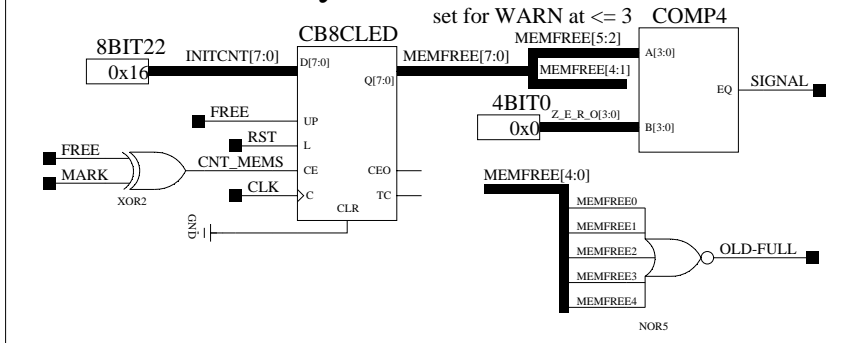
# FIFO Assign Control

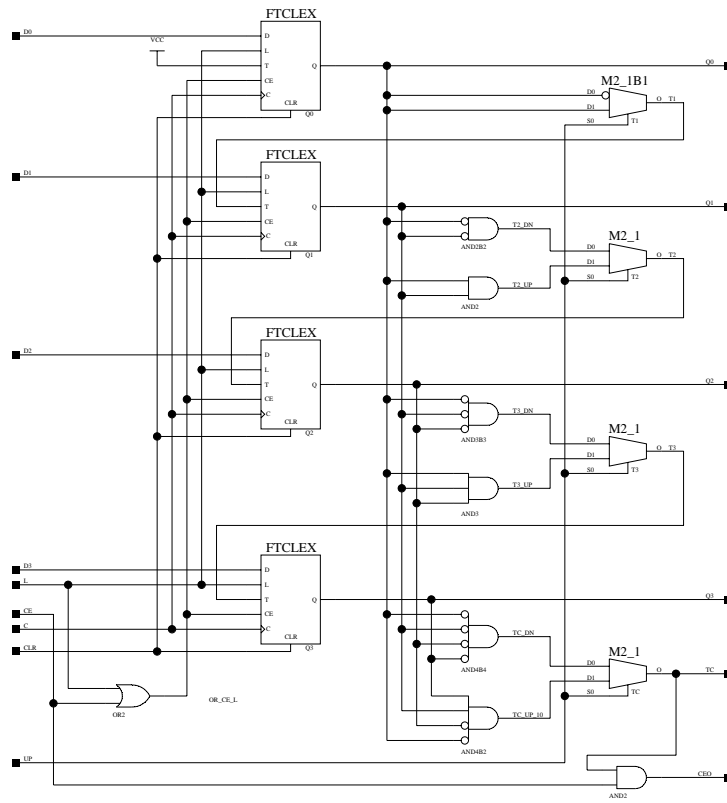


PREF\_C1[1:0]  
BUSY\_C1[1:0]  
GOT\_C1[1:0]



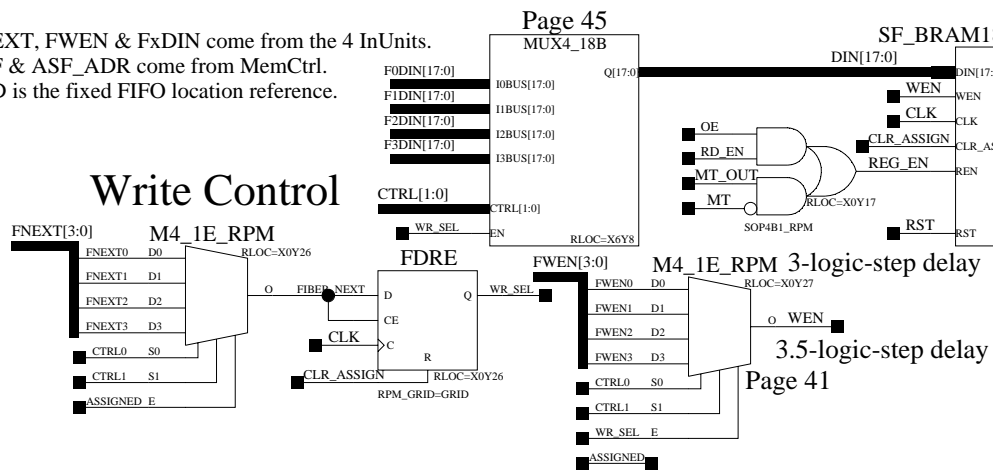
# How many of 22 FIFOs are free?





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

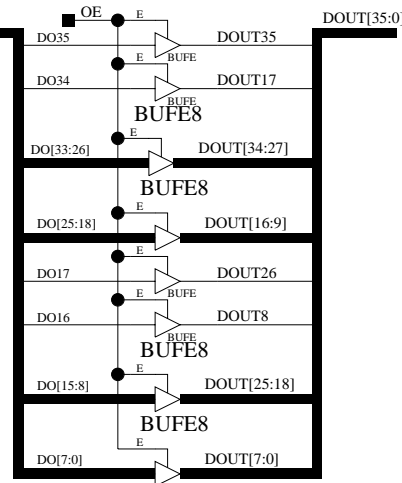
### Write Control



Page 45

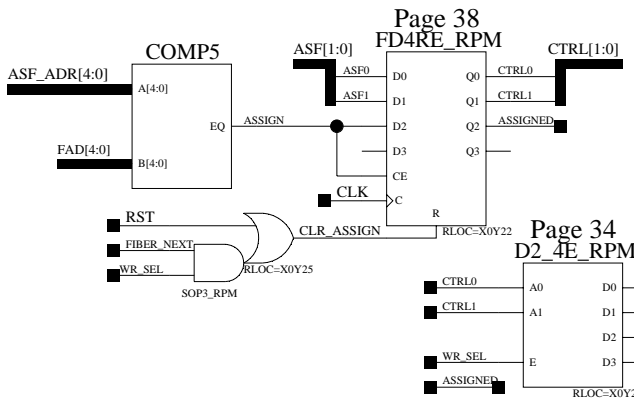
SF\_BRAM18\_36X1024CUSTOM

Page 11



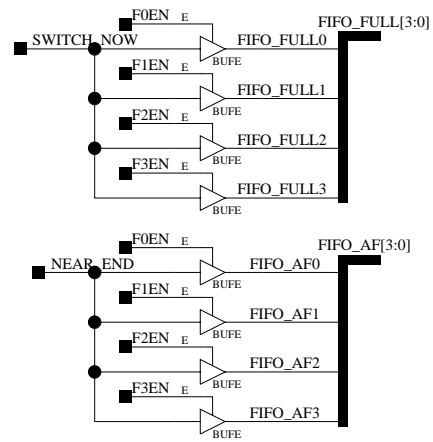
### MemIn-MemOut Mapping:

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

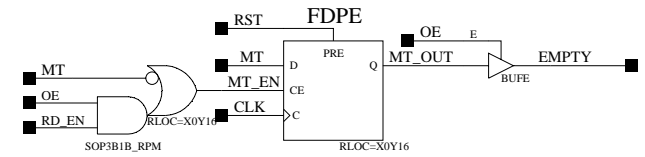


Page 38

Page 34



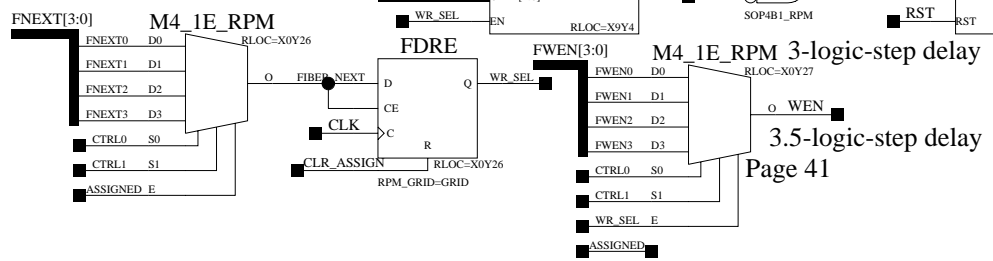
### First Word Fall Through logic:



Not Used:  $REN = OE * RD\_EN$

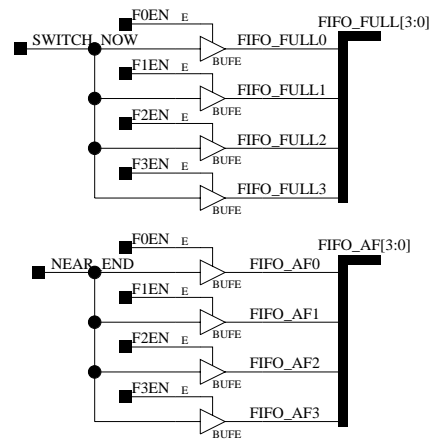
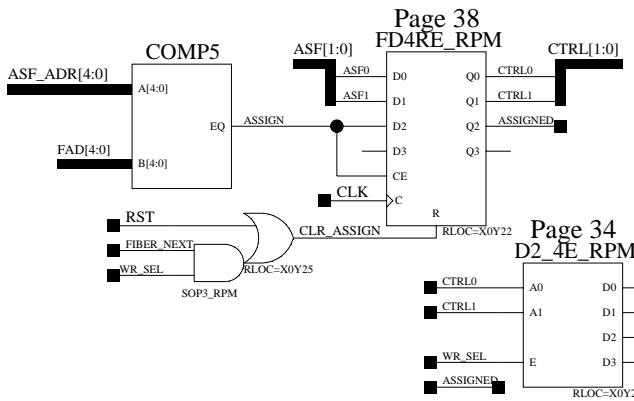
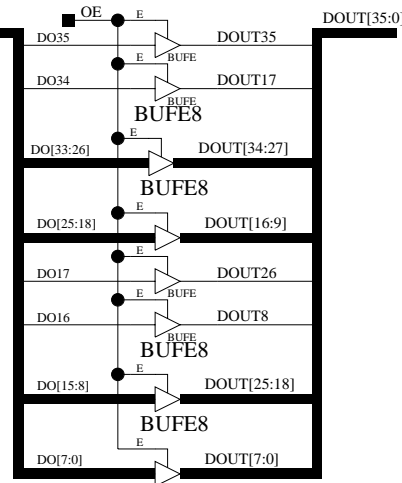
FNEXT, FWEN & FxDIN come from the 4 InUnits.  
 ASF & ASF\_ADR come from MemCtrl.  
 FAD is the fixed FIFO location reference.

### Write Control

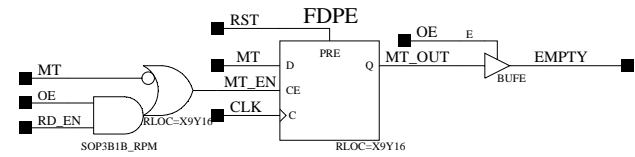


### MemIn-MemOut Mapping:

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:

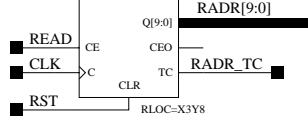


Not Used: REN = OE \* RD\_EN

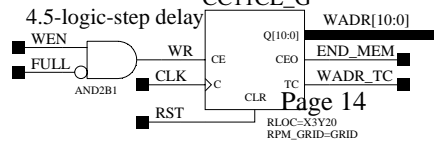


# sfifo18\_36x1024

Page 13  
CC10CE\_G

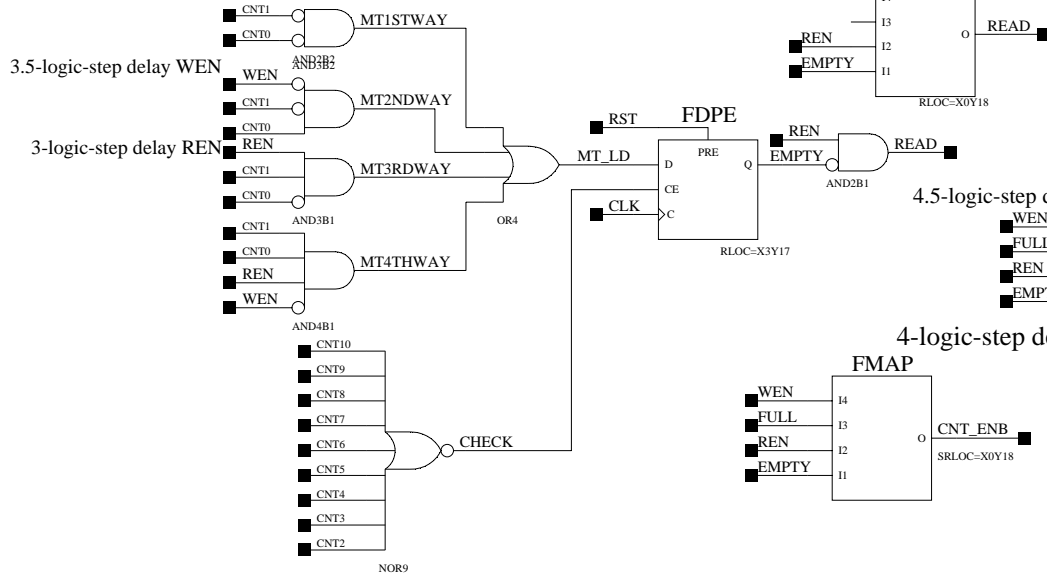


Page 14  
CC11CE\_G

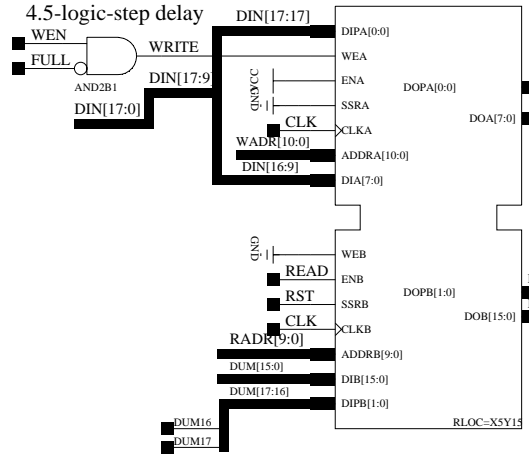


5.5-logic-step delay...

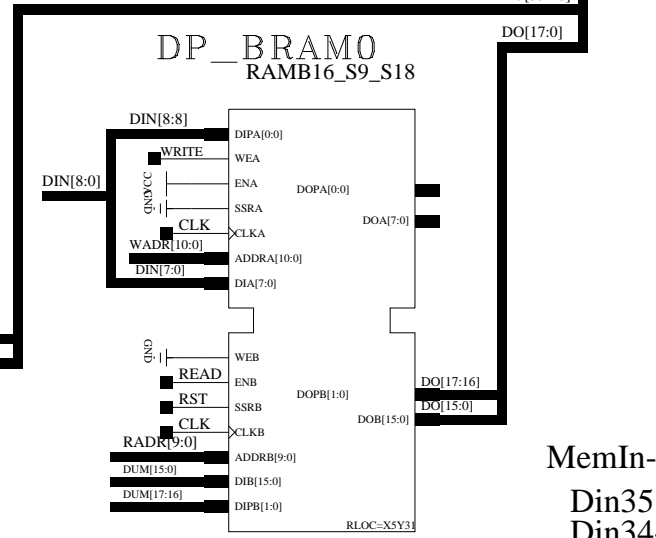
but 4 input logic should optimize to 1 LUT=4.5 step delay.



## DP\_BRAM1



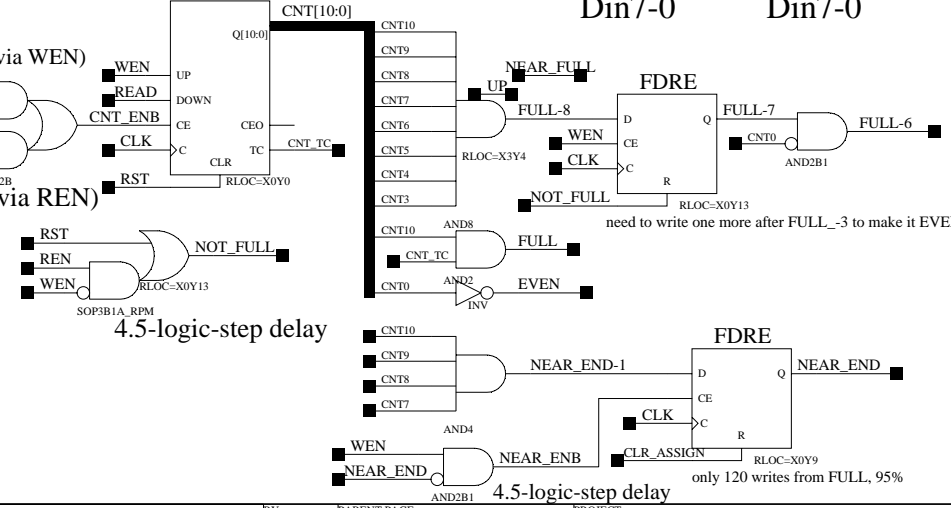
## DP\_BRAM0



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

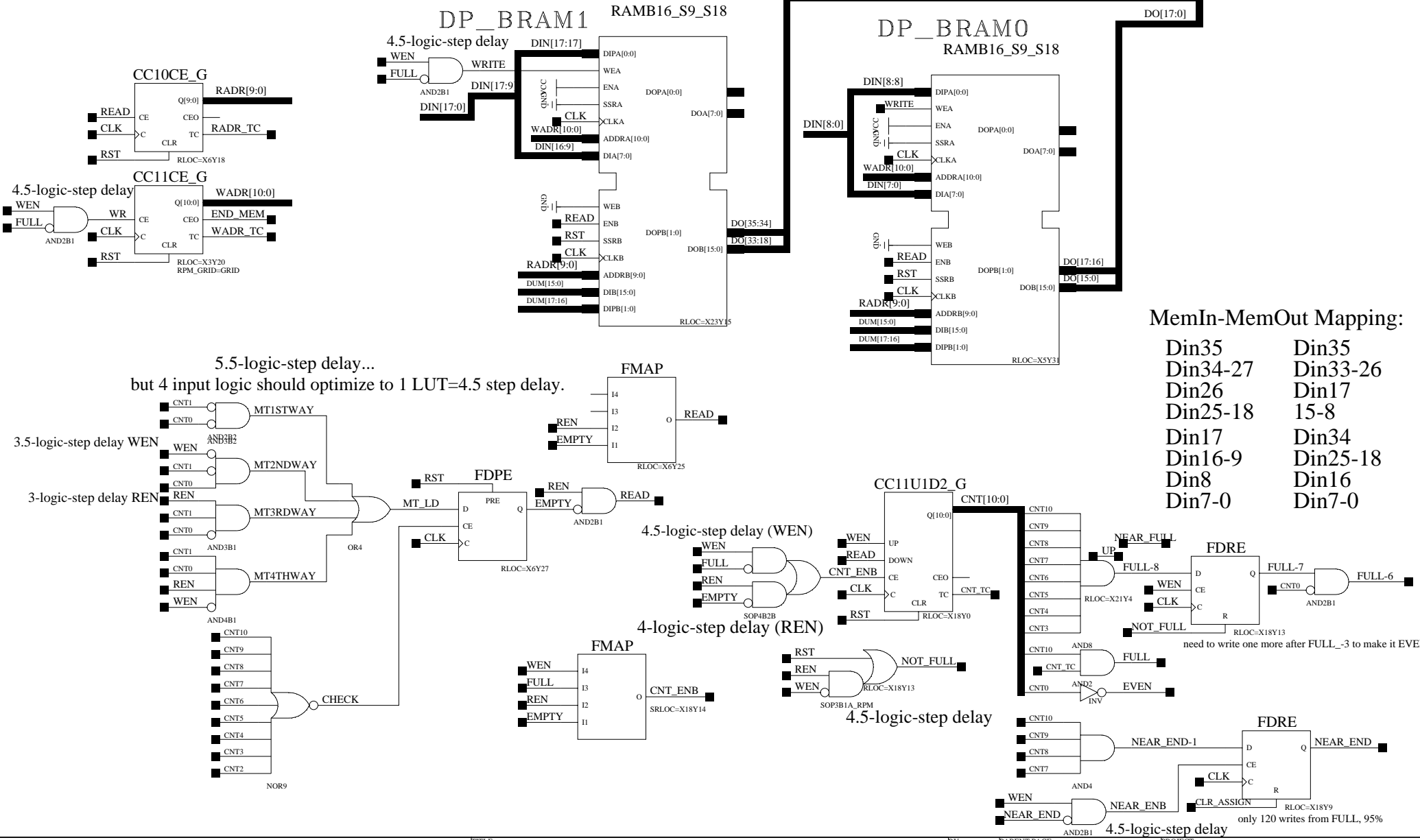
Page 15  
CC11U1D2\_G



need to write one more after FULL\_-3 to make it EVEN

only 120 writes from FULL, 95%

# 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.

3.5-logic-step delay WEN  
3-logic-step delay REN

4.5-logic-step delay (WEN)

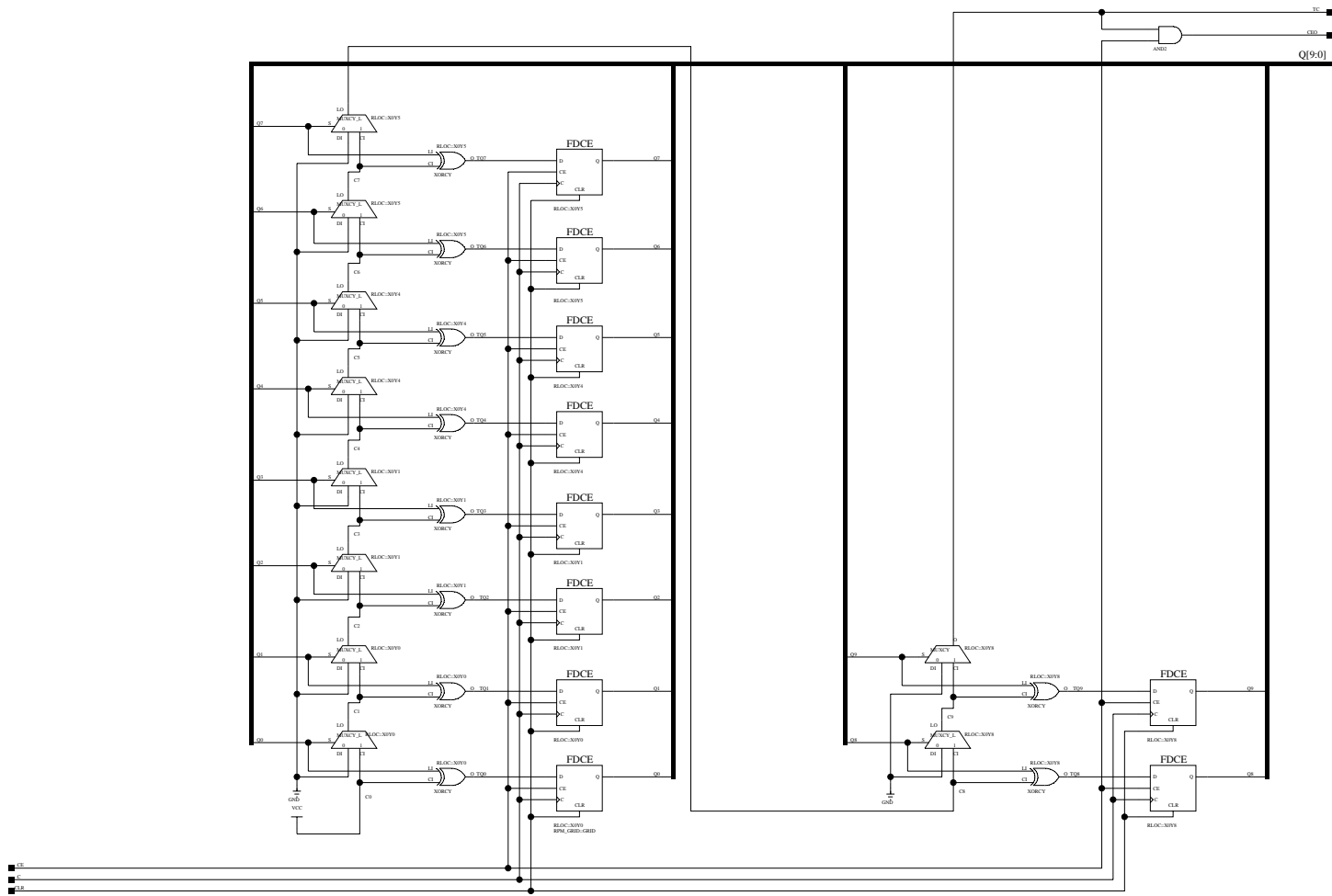
4-logic-step delay (REN)

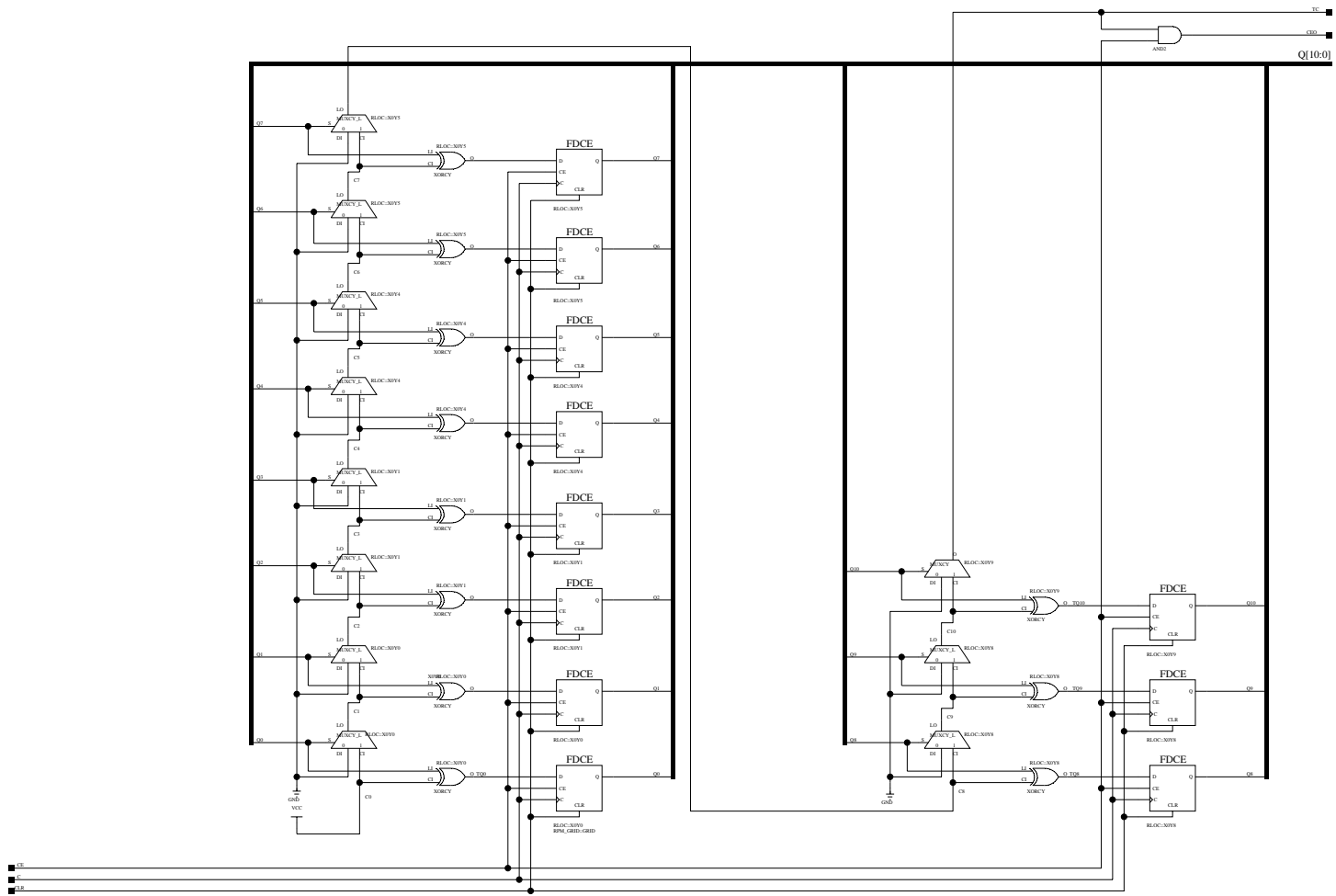
4.5-logic-step delay

4.5-logic-step delay

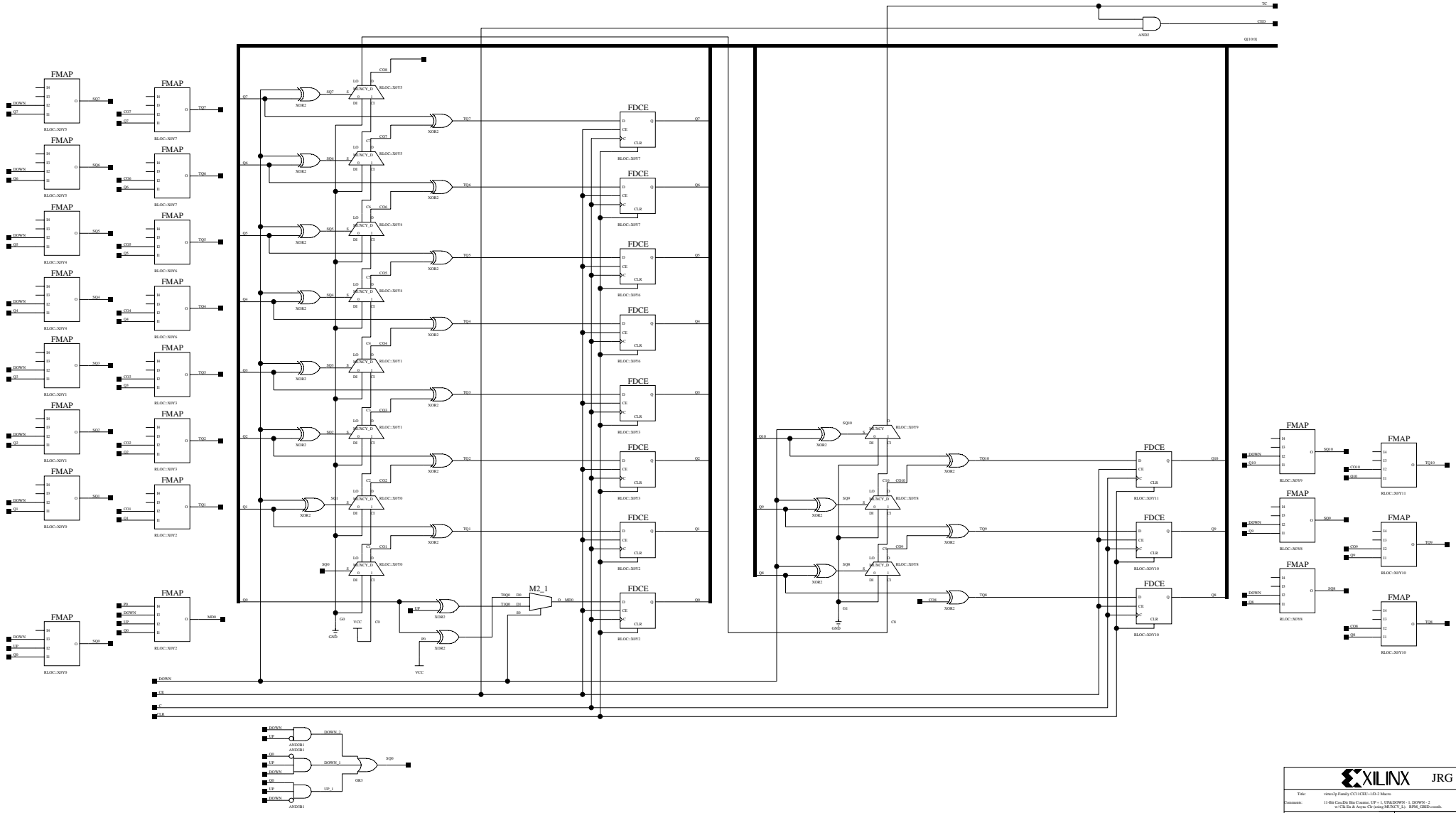
need to write one more after FULL\_-3 to make it EVEN

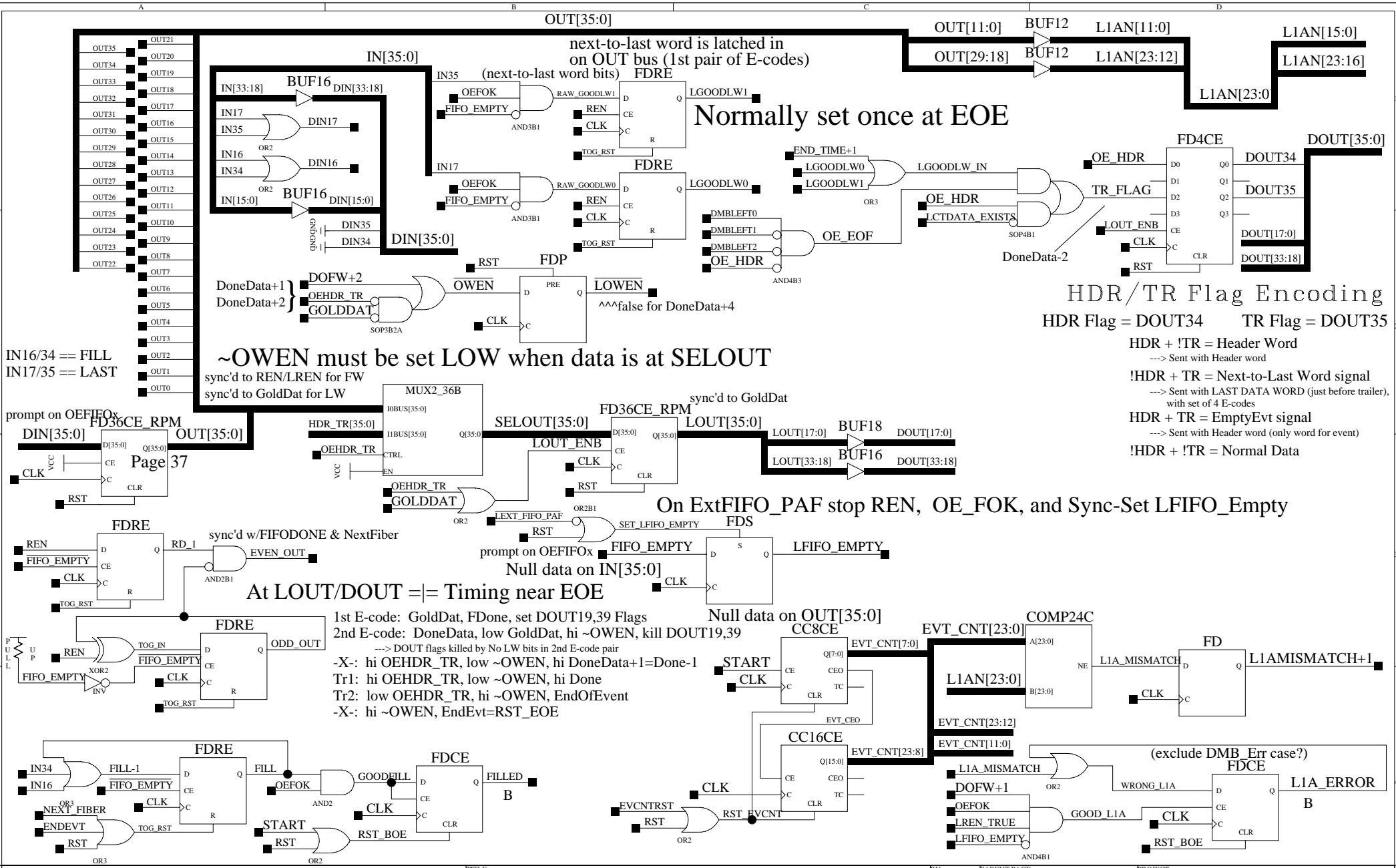
only 120 writes from FULL, 95%





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





IN16/34 == FILL  
IN17/35 == LAST

~OWEN must be set LOW when data is at SELOUT

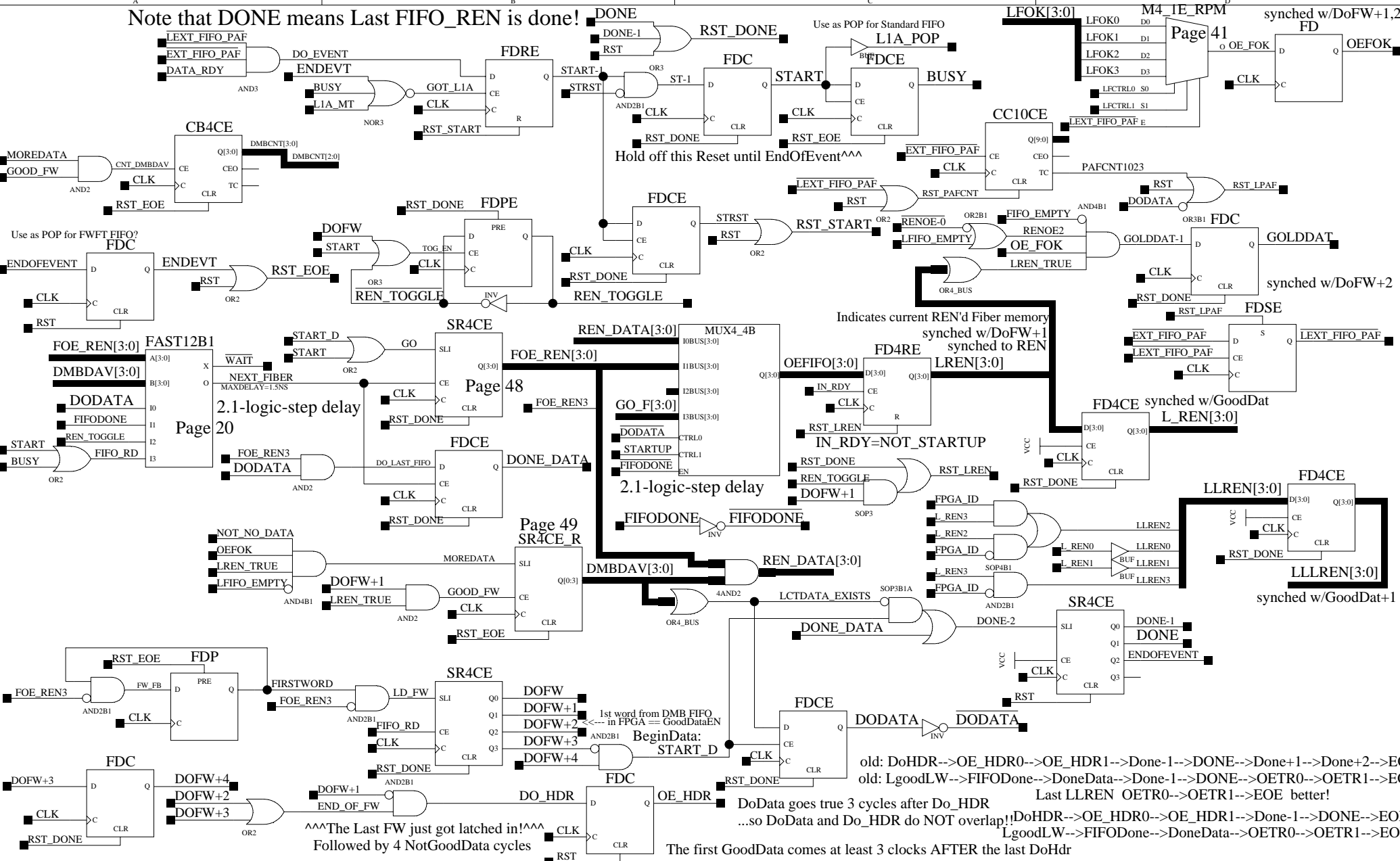
**HDR/TR Flag Encoding**  
 HDR Flag = DOUT34    TR Flag = DOUT35  
 HDR + !TR = Header Word  
 --> Sent with Header word  
 !HDR + TR = Next-to-Last Word signal  
 --> Sent with LAST DATA WORD (just before trailer),  
 with set of 4 E-codes  
 HDR + TR = EmptyEvt signal  
 --> Sent with Header word (only word for event)  
 !HDR + !TR = Normal Data

At LOU/DOUT == Timing near EOE

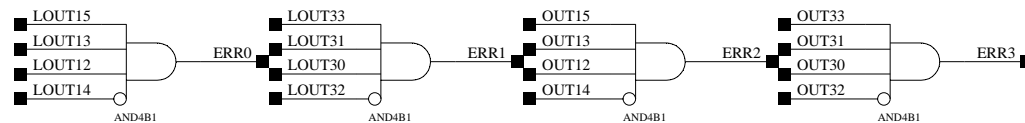
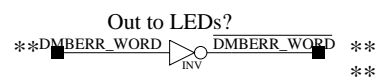
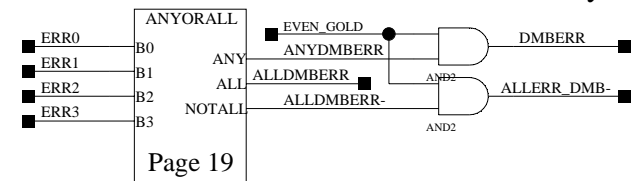
1st E-code: GoldDat, FDone, set DOUT19,39 Flags  
 2nd E-code: DoneData, low GoldDat, hi ~OWEN, kill DOUT19,39  
 --> DOUT flags killed by No LW bits in 2nd E-code pair  
 -X-: hi OEHDR\_TR, low ~OWEN, hi DoneData+1=Done-1  
 Tr1: hi OEHDR\_TR, low ~OWEN, hi Done  
 Tr2: low OEHDR\_TR, hi ~OWEN, EndOfEvent  
 -X-: hi ~OWEN, EndEvt=RST\_EOE

On ExtFIFO\_PAF stop REN, OE\_FOK, and Sync-Set LFIFO\_Empty

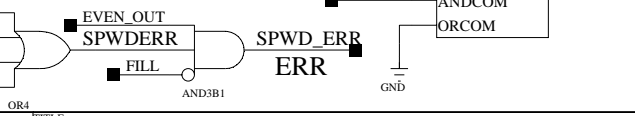
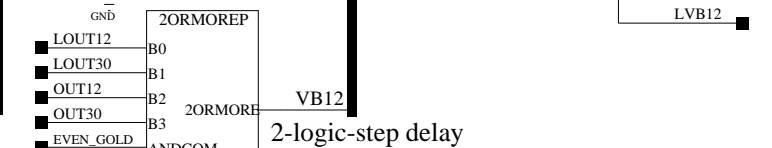
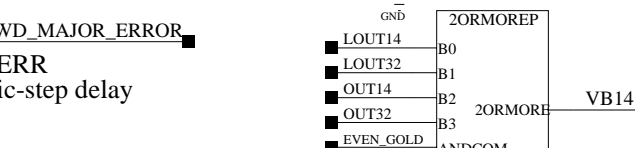
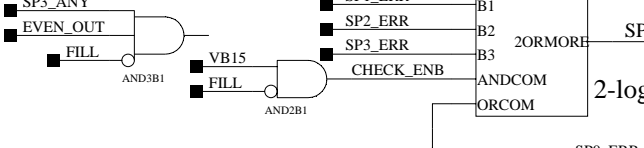
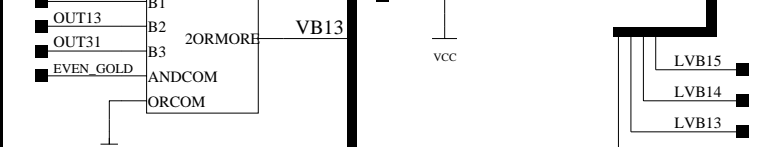
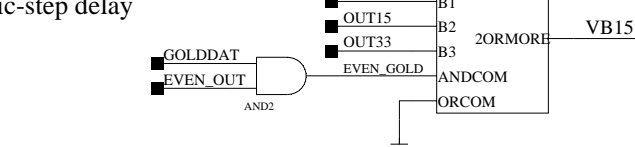
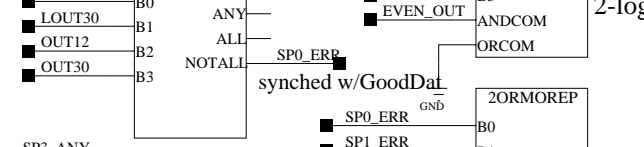
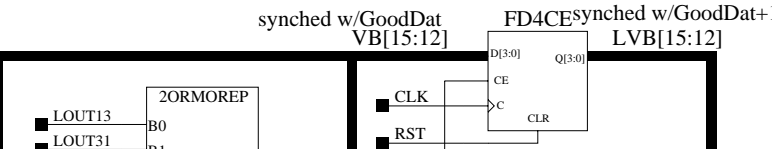
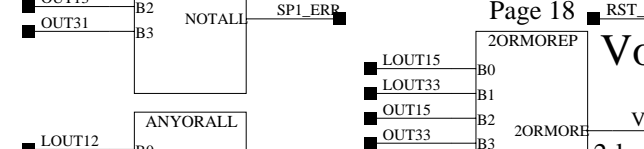
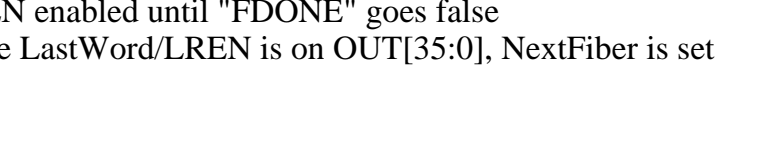
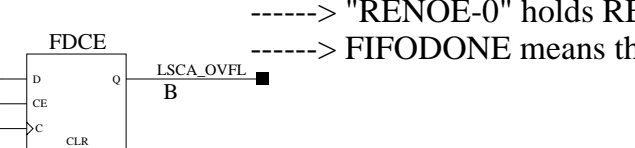
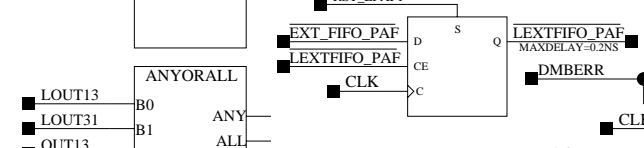
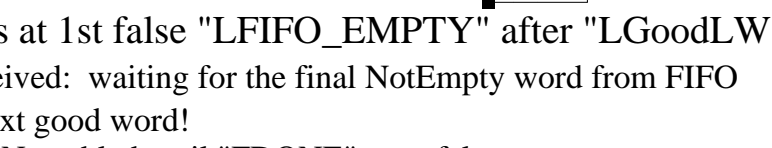
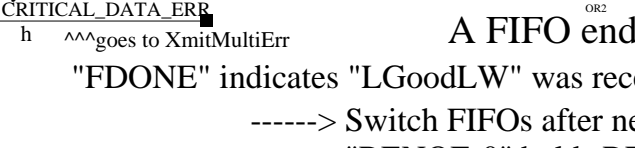
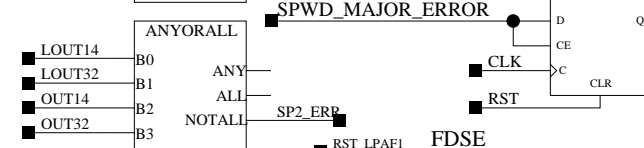
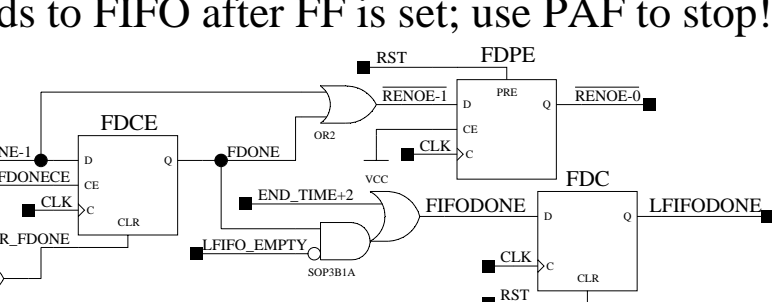
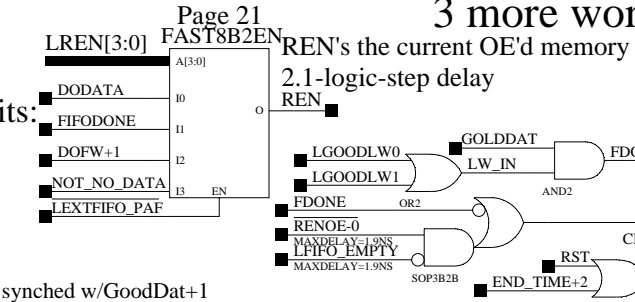
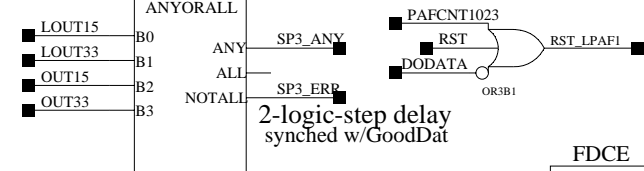
# 3 more words to FIFO after FF is set; use LEXT\_FIFO\_PAF to stop REN/OWEN!



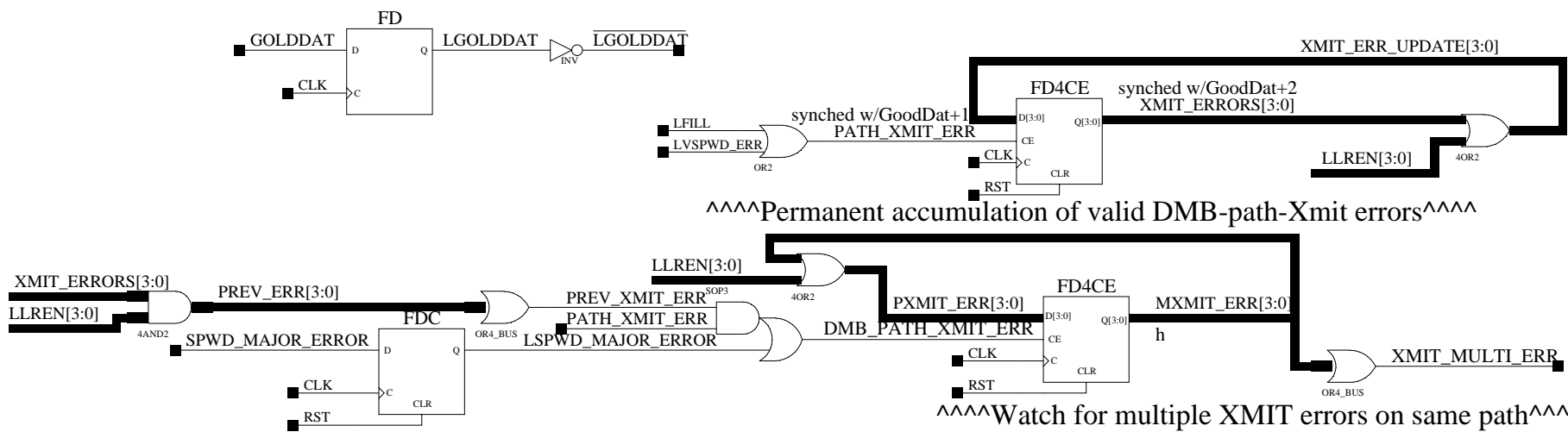
Check for DMB Error Word and consistency:



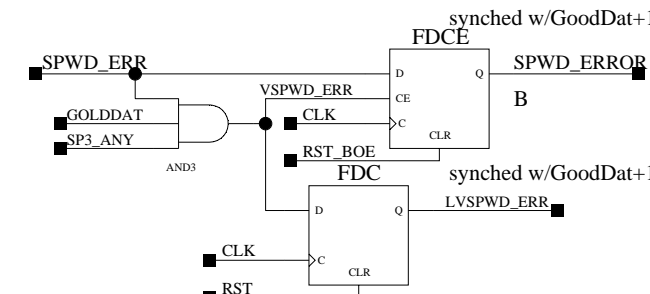
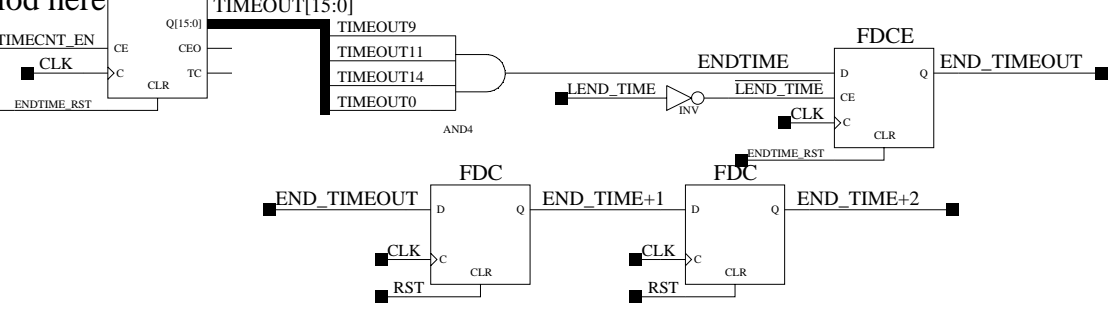
Check consistency of the four "Special Word" bits:







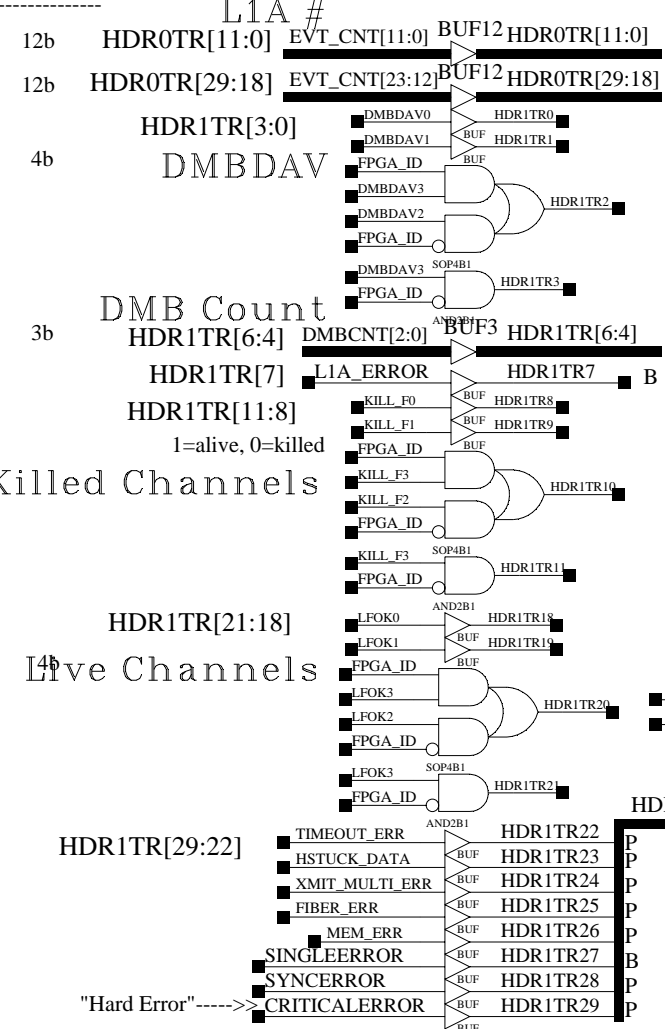
FIFO Done Timeout: count to 10752, 132 usec=10625 is the worst case possible per FIFO?  
 Set to 18945 (~236 usec) for now



# Status Monitor

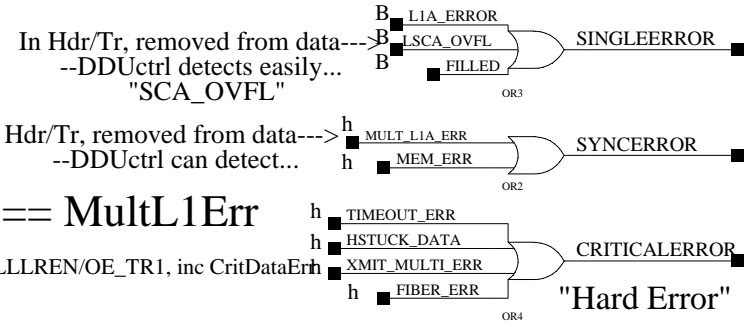
Pick 8 Status signals, include in DAQ HDR/TR  
 -create logic for OEHDR/TR and their format  
 Also set 4 FMM bits for each input channel

#Bits useful at EOE (for Used Memories in event? 22 bits)

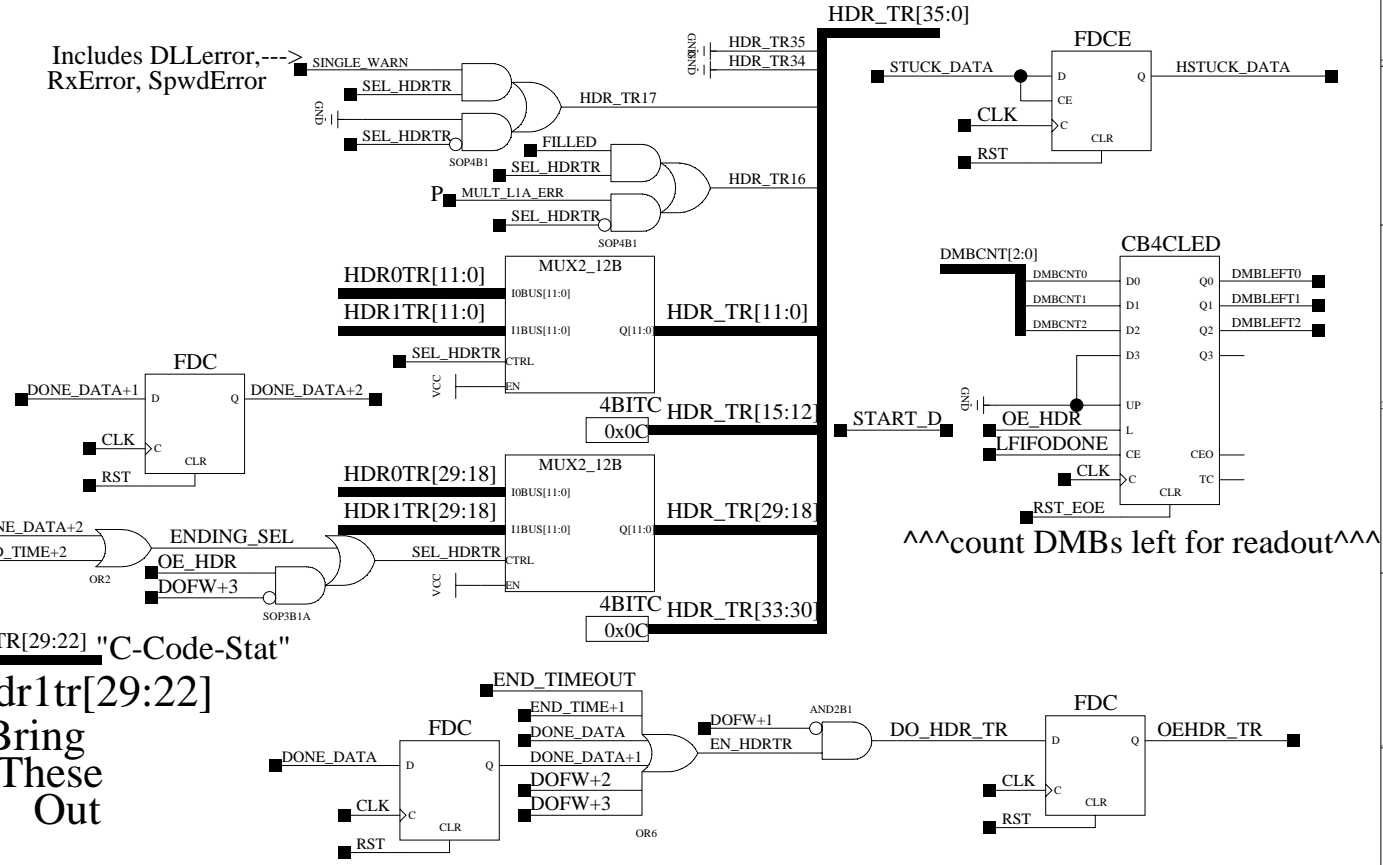


SyncErr AND ~MemErr == MultL1Err

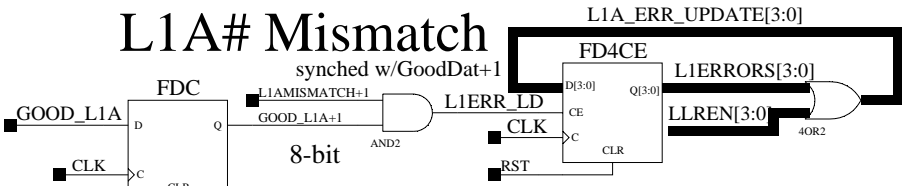
synced to LLLLREN/OE\_TR1, inc CritDataErh



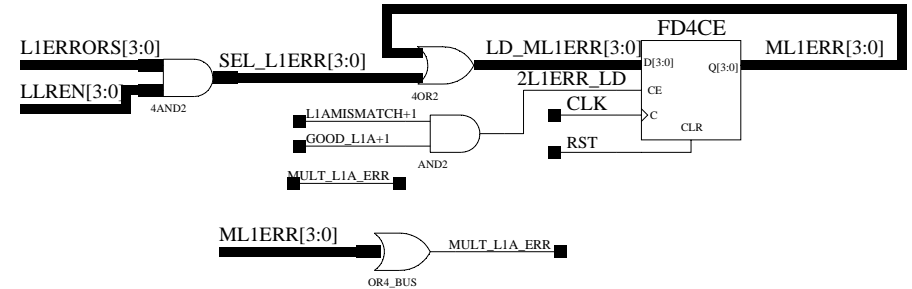
Includes DLLerror, RxError, SpwdError



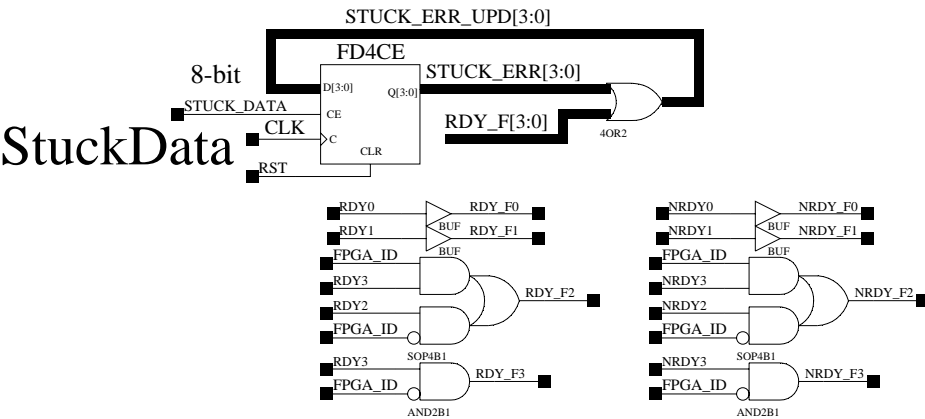
# L1A# Mismatch



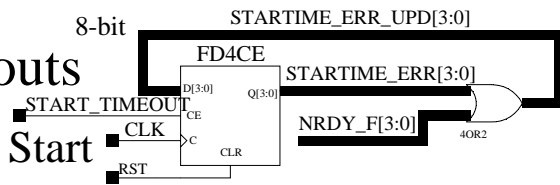
# Multi-L1A# Mismatch



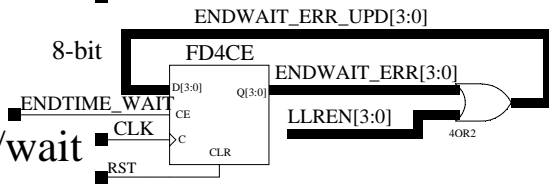
# StuckData



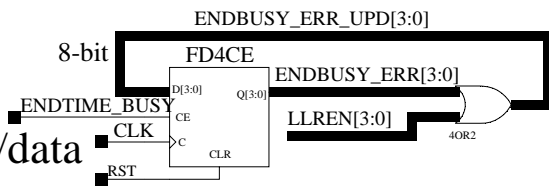
# Timeouts



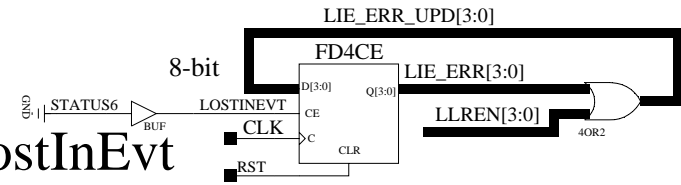
# End w/wait



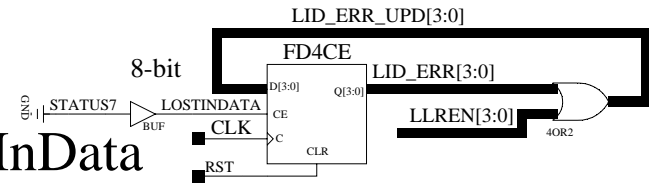
# End w/data



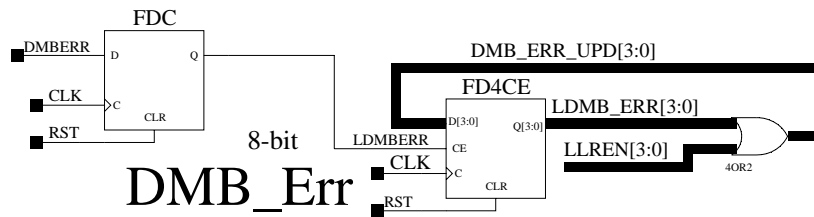
# LostInEvt



# LostInData



# DMB\_Err



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

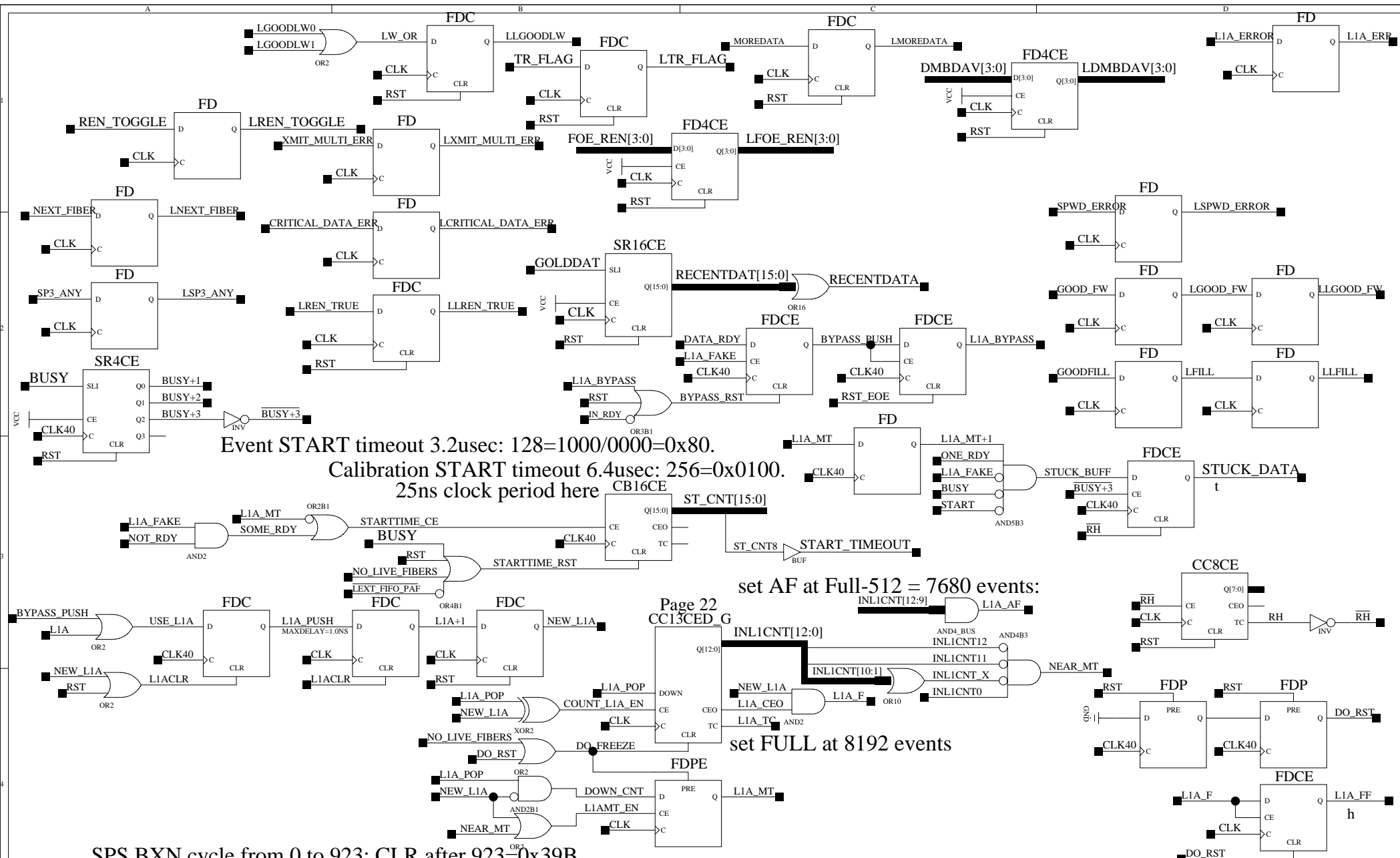
H2: 0x/8000/0001/8000/HHHH

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:

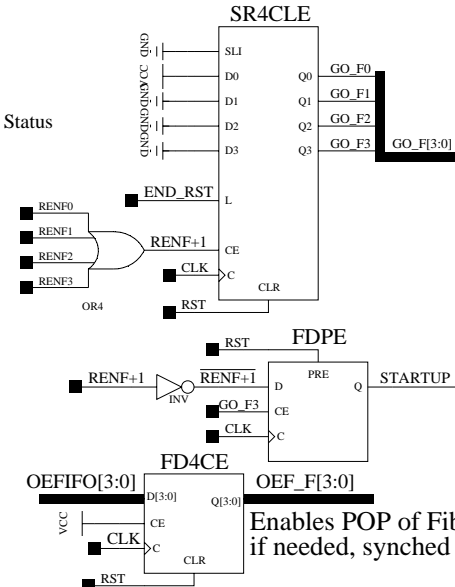
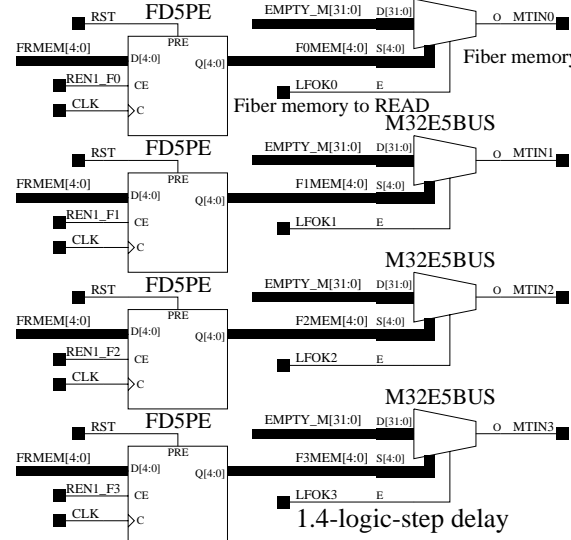
set FULL at 8192 events

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

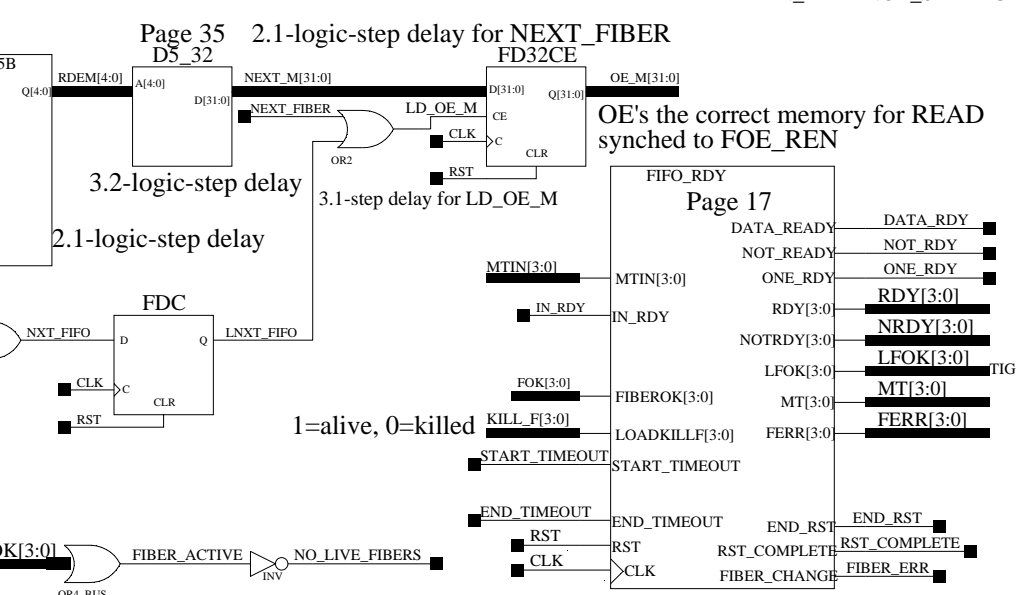
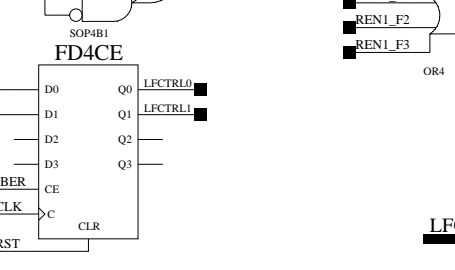
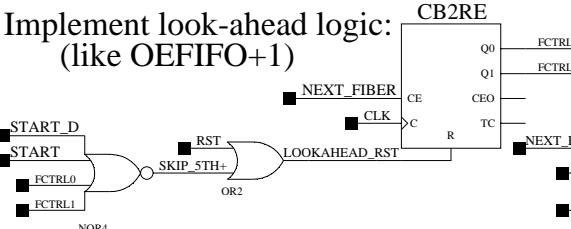
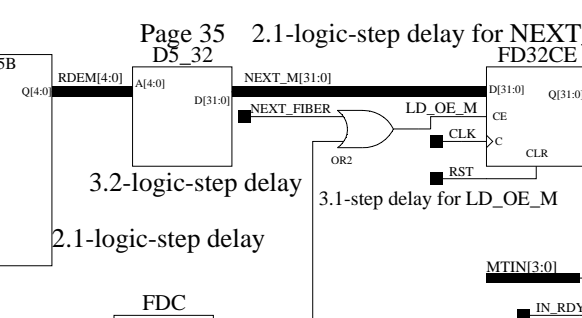
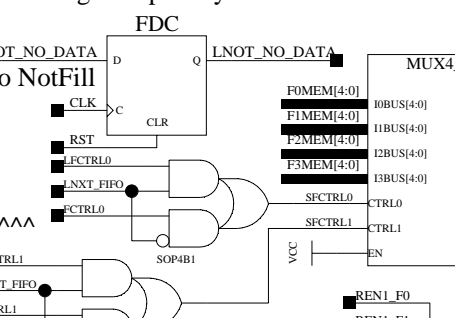
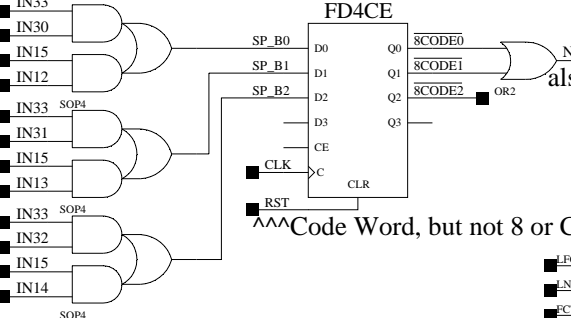
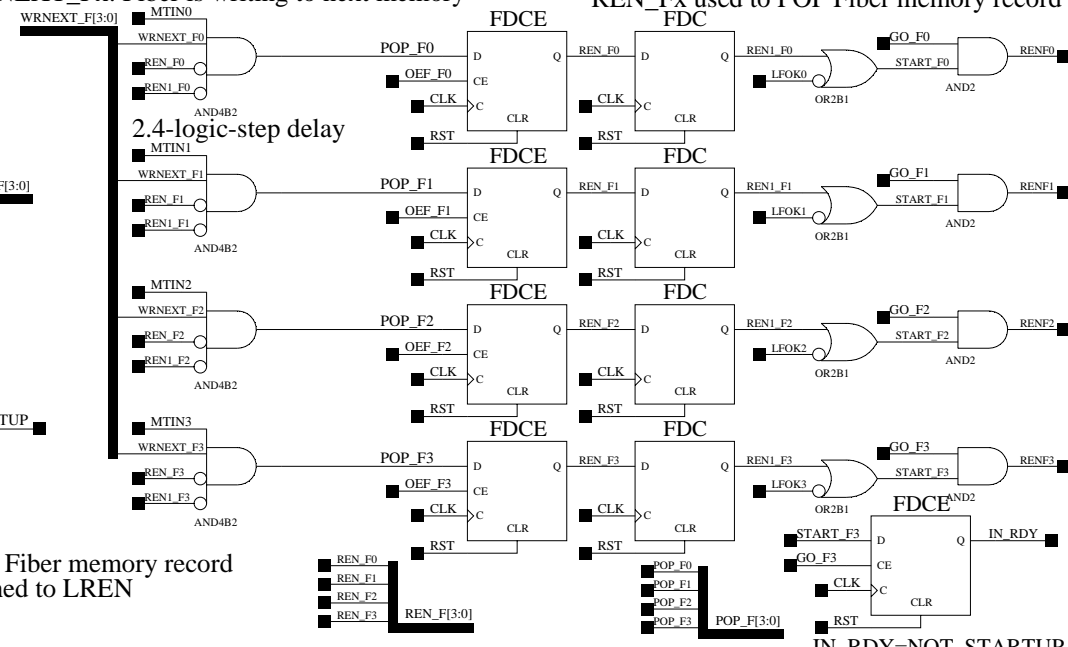
Page 22  
 CC15CED G

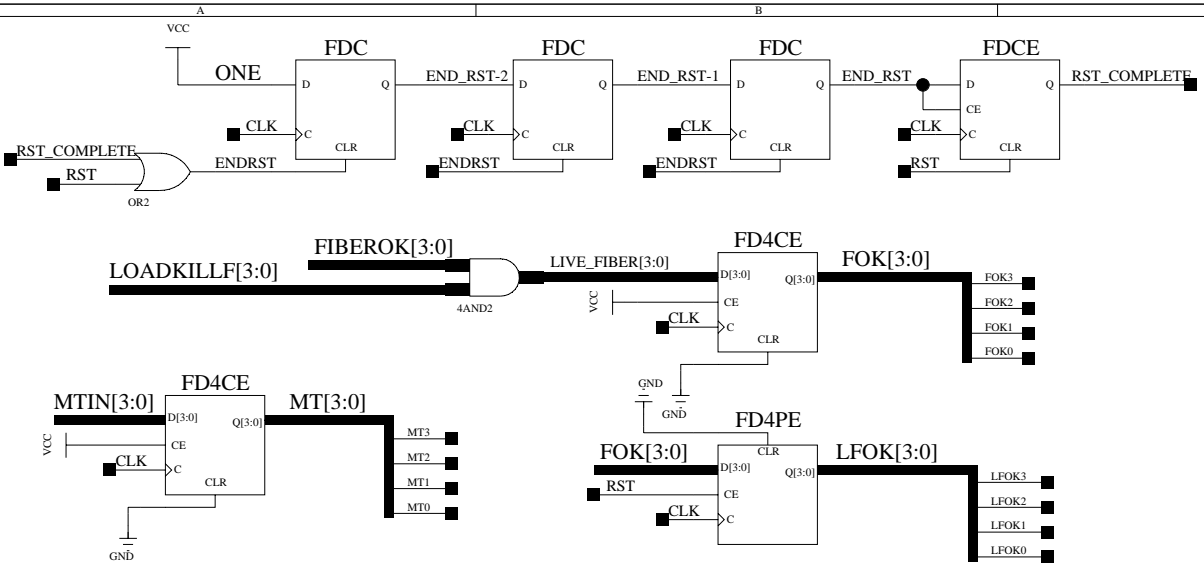
# END

Page 25  
M32E5BUS

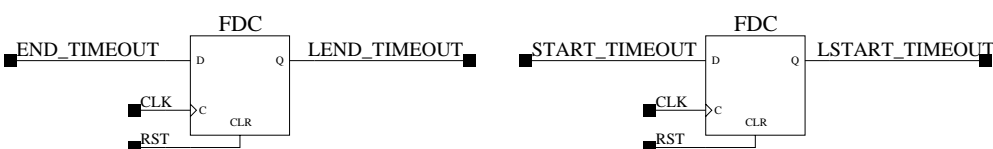


WRNEXT\_Fx: Fiber is writing to next memory

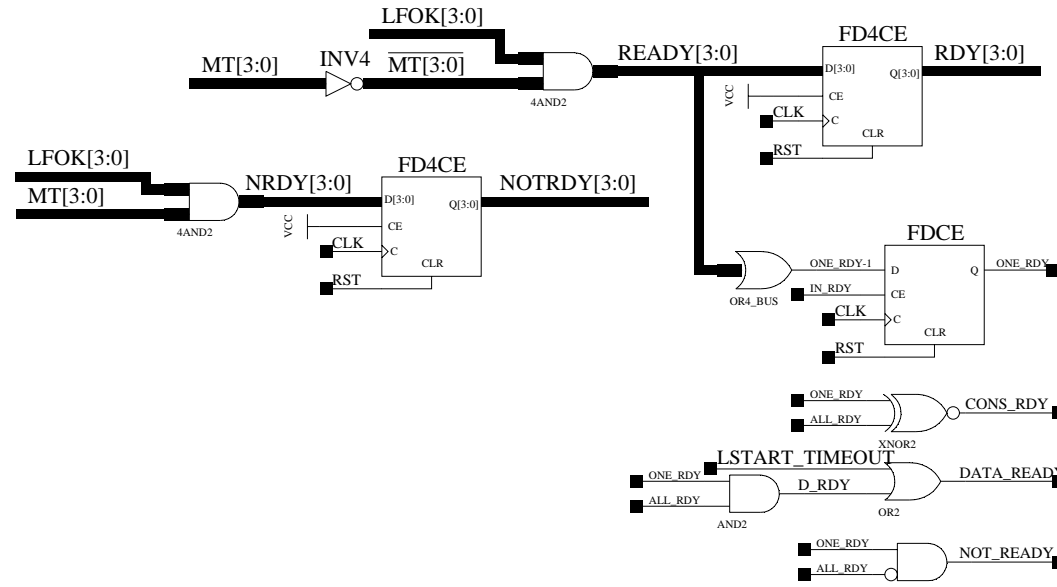


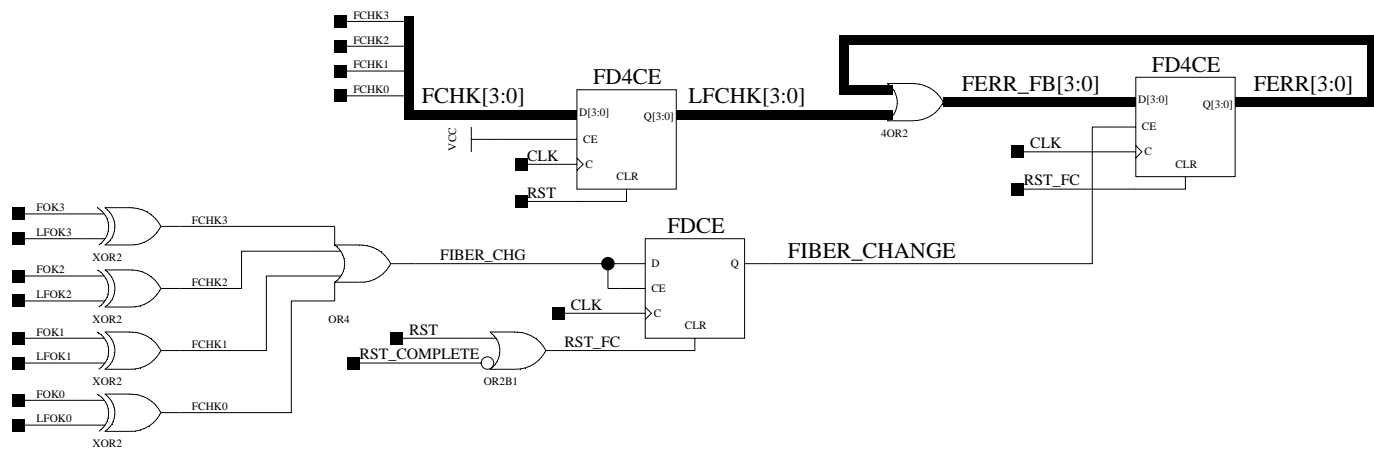


Access FIBEROK from JTAG as a fiber check.  
 ---> RESET required after fiber status change for now...  
 Change in FIBEROK is BAD! Set error code.  
 ...notify FMM and maybe set STATUS bit, but  
 ...data is OK until FIFO read time-out occurs.  
 ...but how to know WHEN the bad-data comes out?  
 --->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

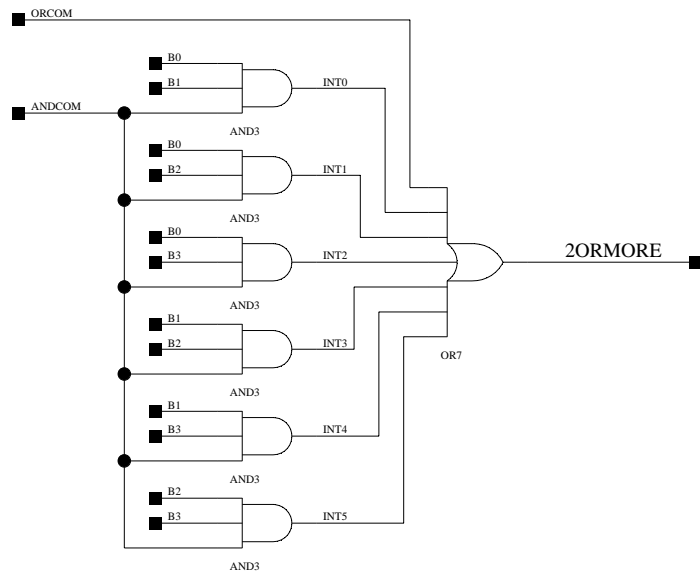


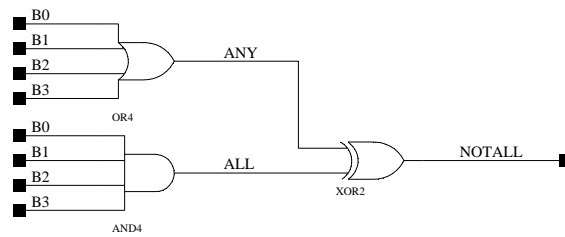
MASK fiber on Start/End TIMEOUT? Kill it in LFOK...

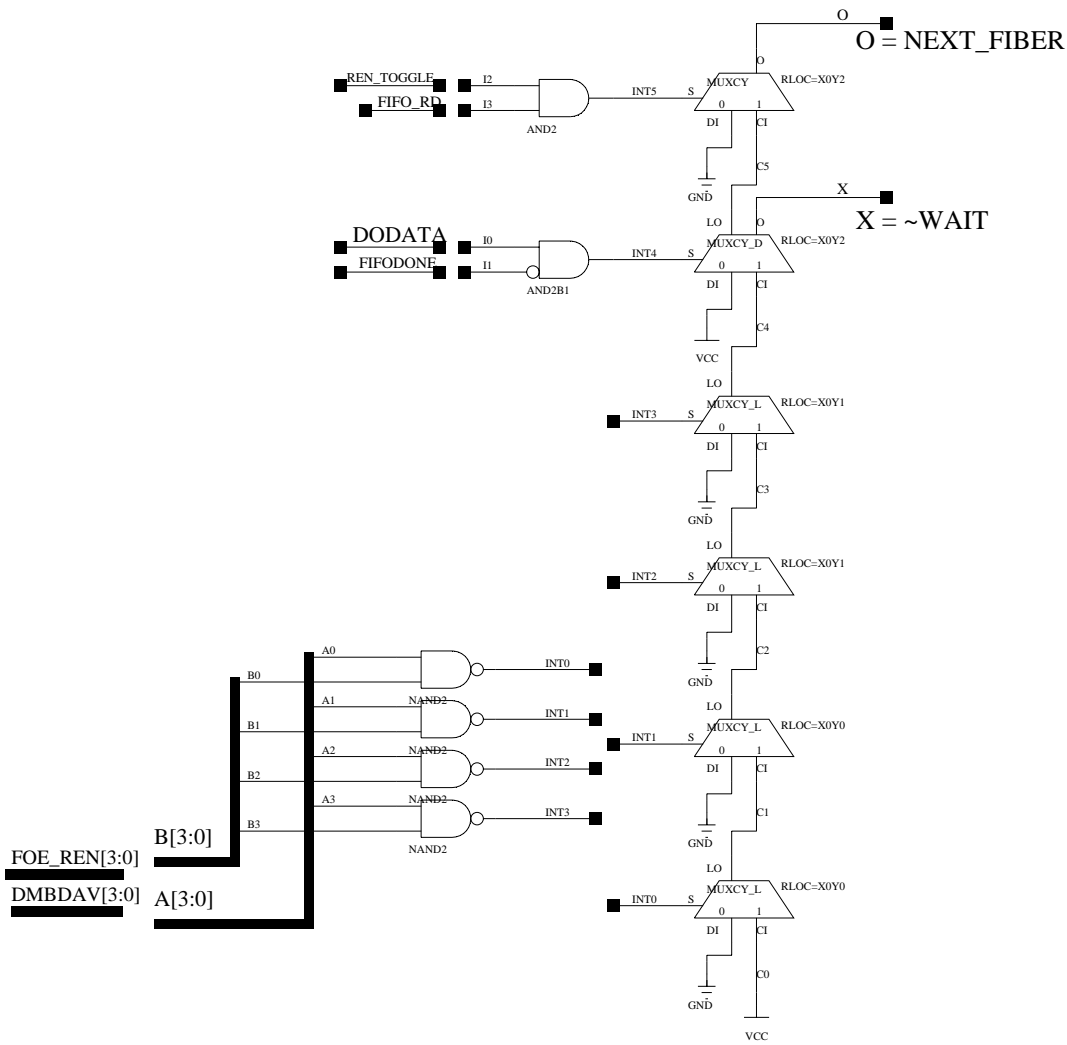




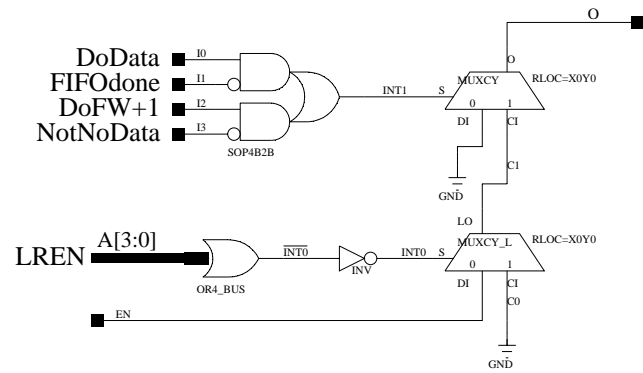








Title:	FAST12B1	
Comments:	Custom Fast, Complex Logic for DDU, use 4 MUXCY as OR, 2 as AND similar to: OR of 4 AND2, AND, AND2B1, AND, AND2	
Date:	15th October 2003	Ver: 1
Sheet Size:	B	Rev: A

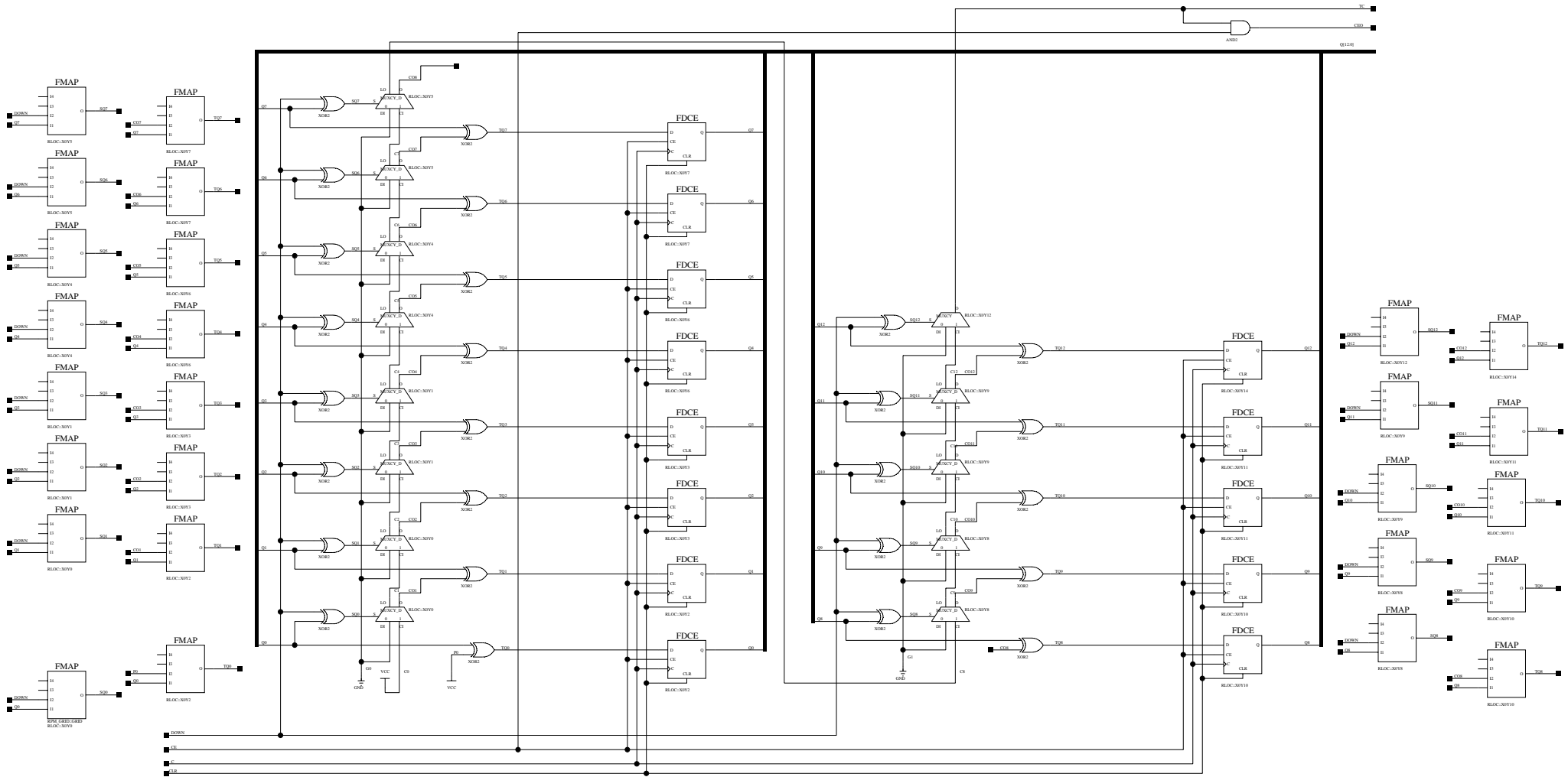


JRG

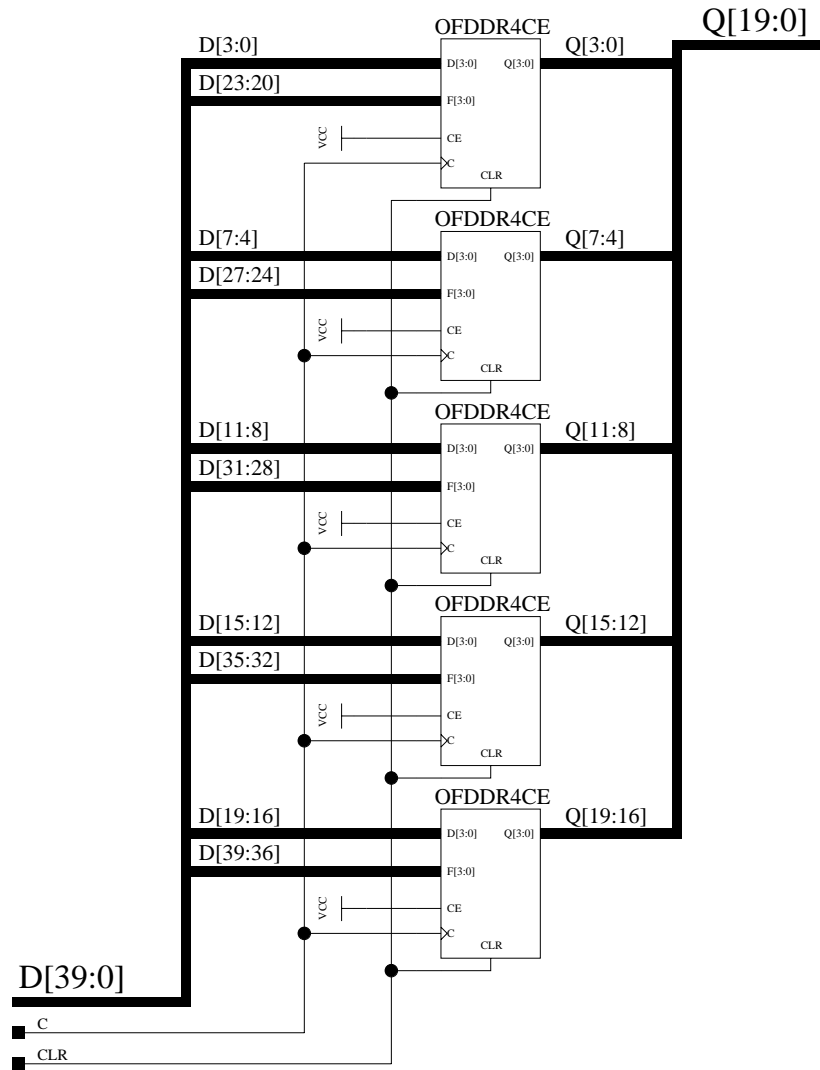
Title:	FAST8B2EN	
Comments:	Custom Fast, Complex Logic for DDU, use 2 MUXCY as AND similar to: OR4 AND SOP4B2b	
Date:	14th September 2004	Ver: 1
Sheet Size: B		Rev: A

DOWN = Decrement 1

Assume that CE includes DOWN+UP

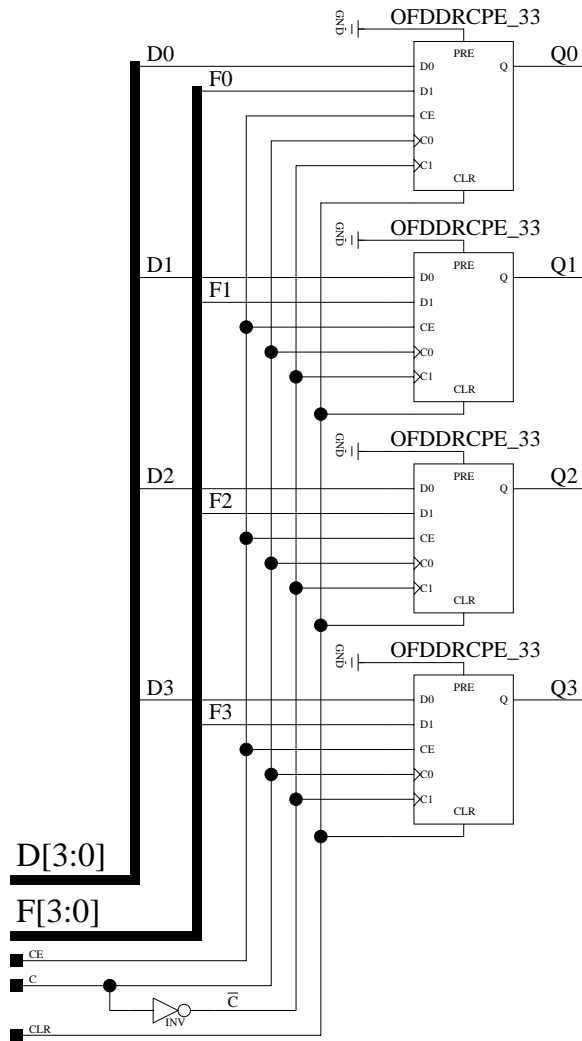


the D byte goes out last (on C rising edge)  
 the F byte goes out first (on C falling edge)

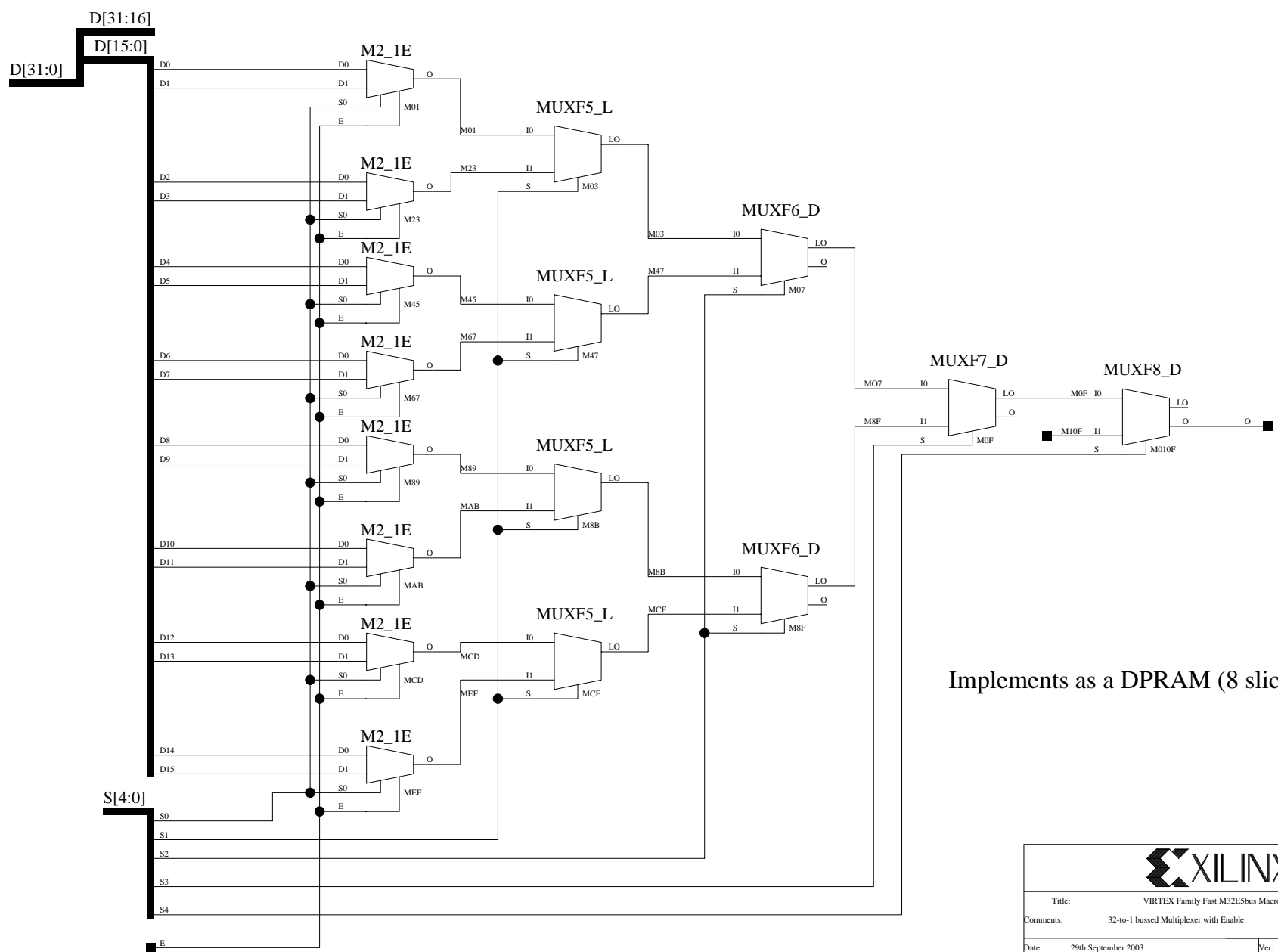


the D byte goes out last (on C rising edge)  
 the F byte goes out first (on C falling edge)

Q[3:0]



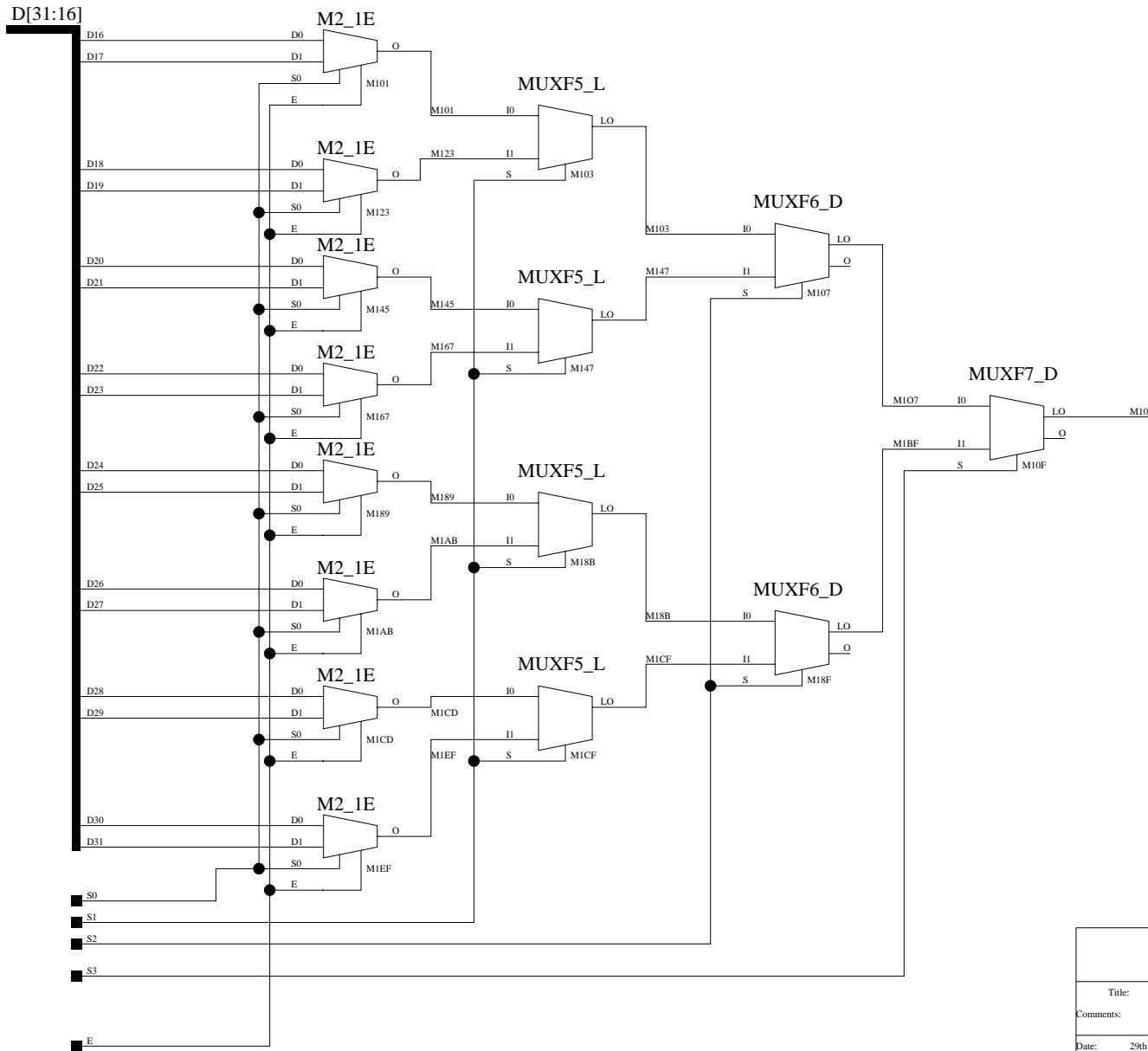
Title: VIRTEX Family OFDDR4 Macro	
JRG	
Comments: 4-Bit Output DDR Flip-Flop w/Asynchronous Clear and Clock Enable	
Date: 16th December 2003	Ver: 1
Sheet Size: B	Rev: A




Implements as a DPRAM (8 slices)

		JRG
Title:	VIRTEX Family Fast M32ESbus Macro	
Comments:	32-to-1 bussed Multiplexer with Enable	
Date:	29th September 2003	Ver: 1
Sheet Size:	B	Rev: A



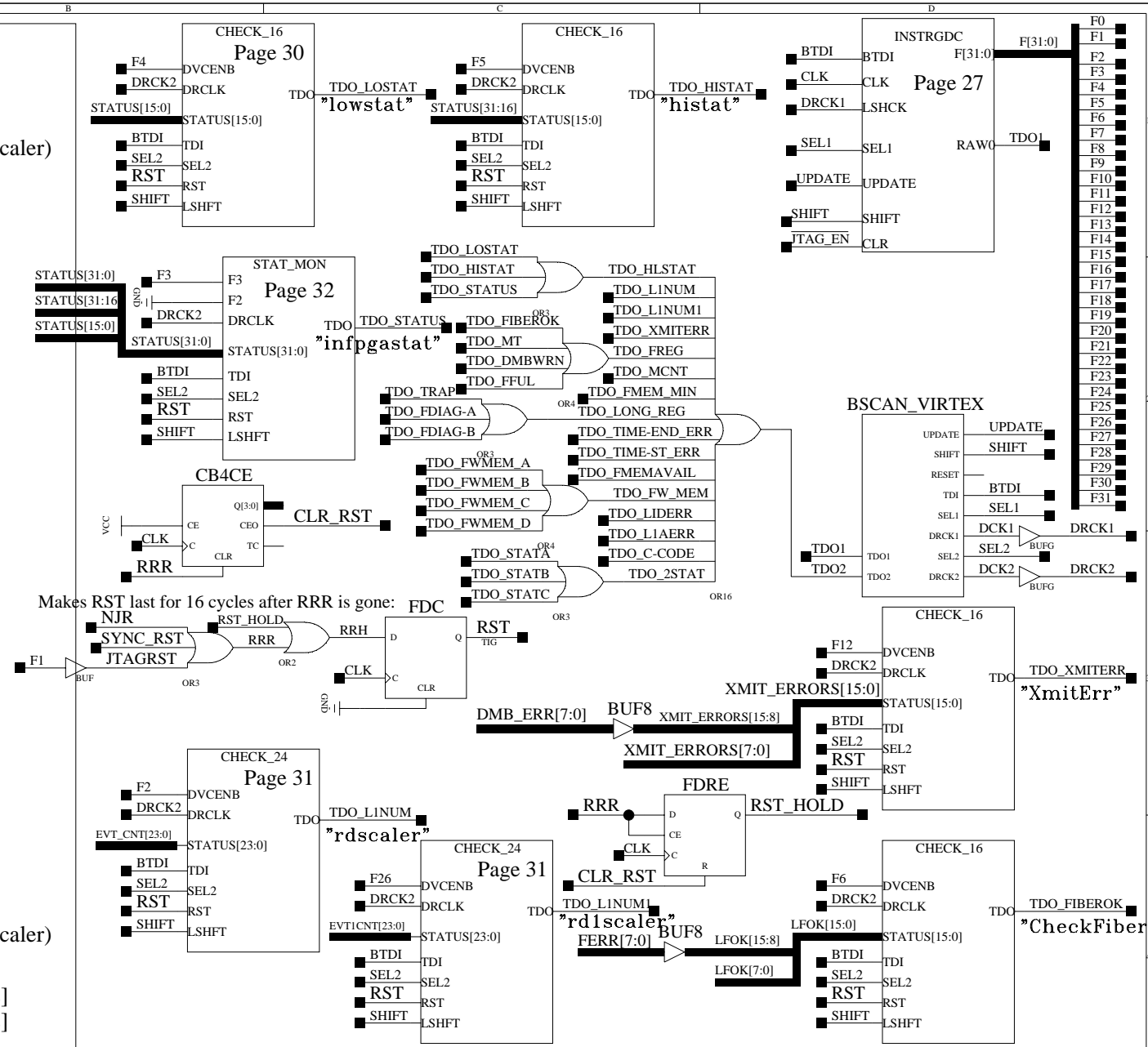


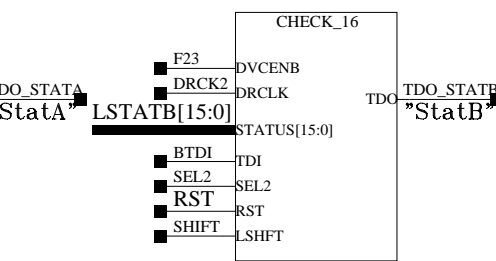
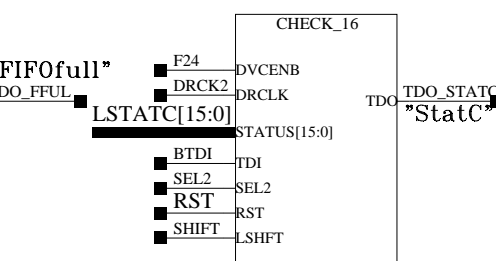
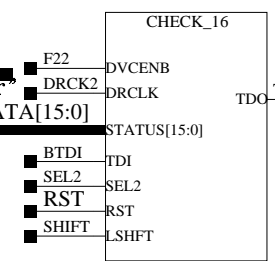
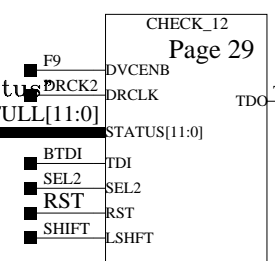
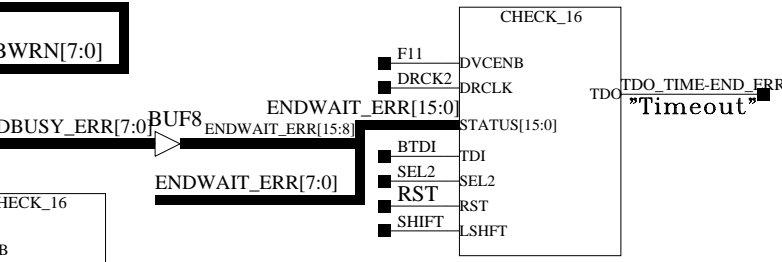
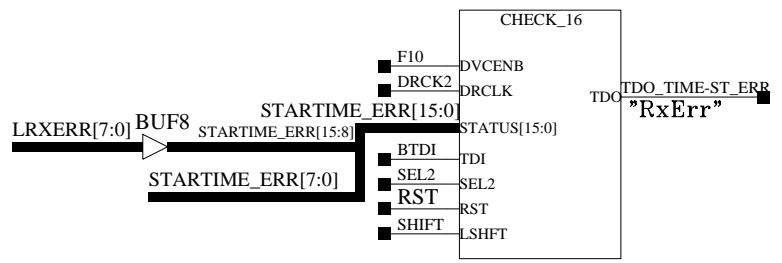
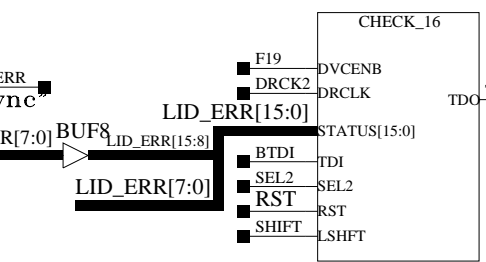
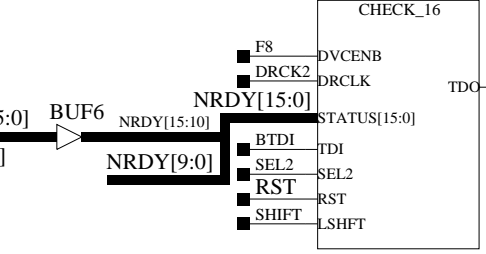
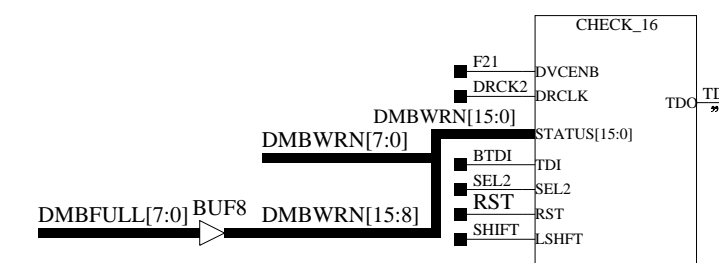
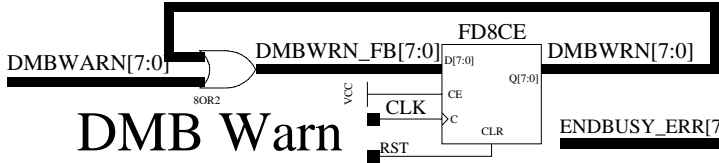
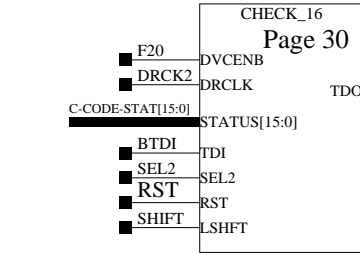
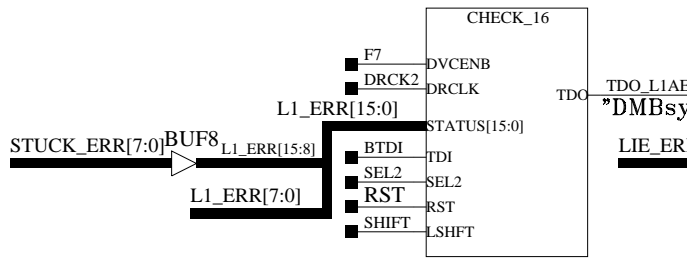
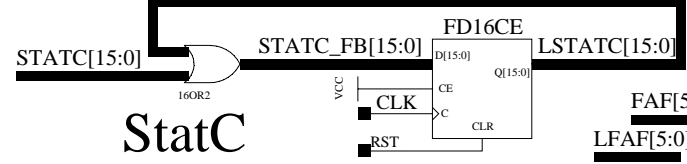
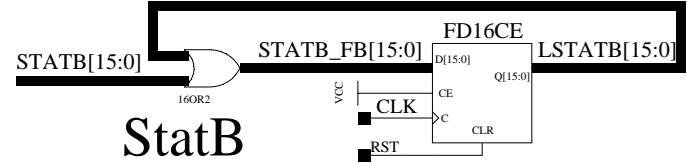
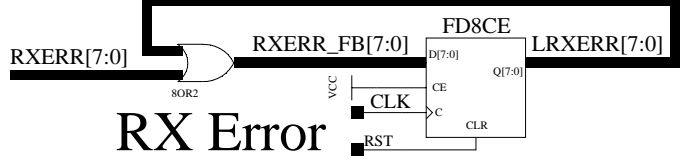
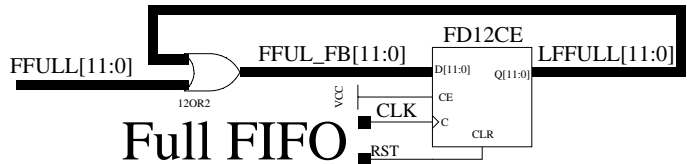
drawn by KS  
Copyright (c) 1993, Xilinx Inc.

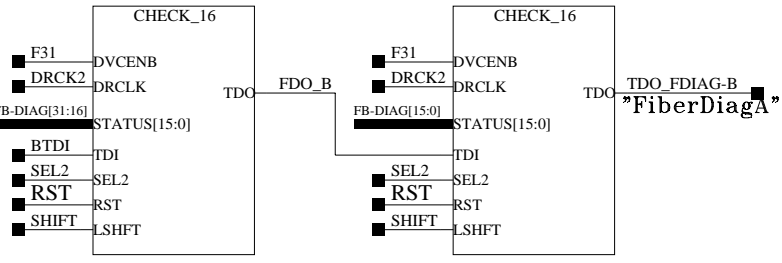
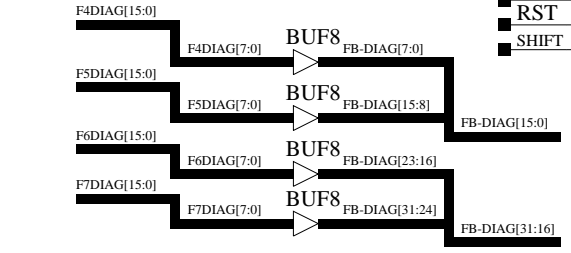
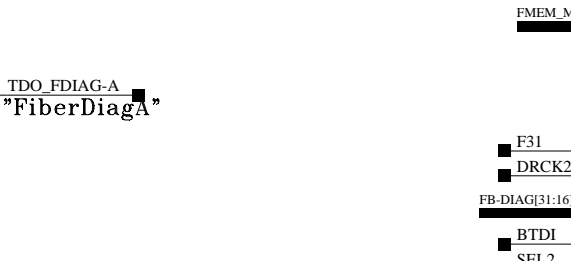
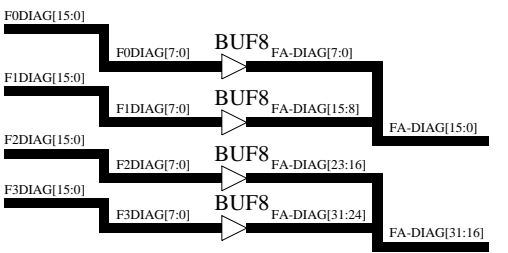
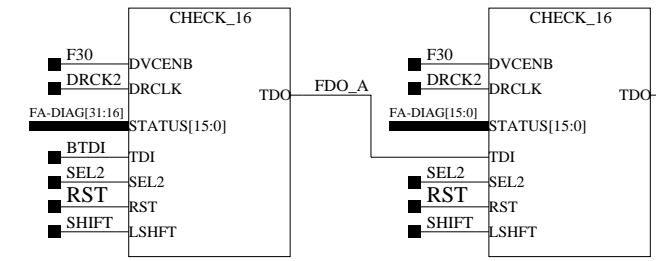
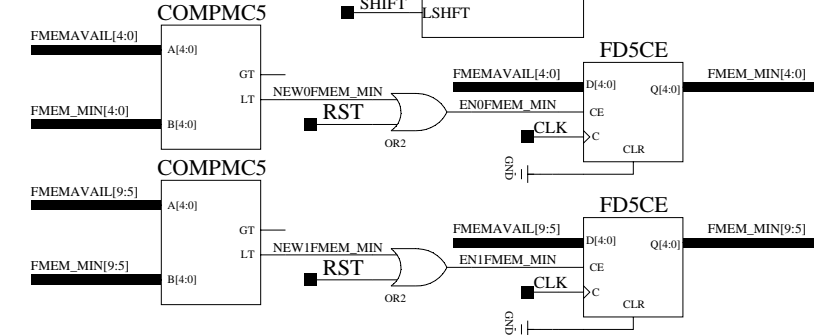
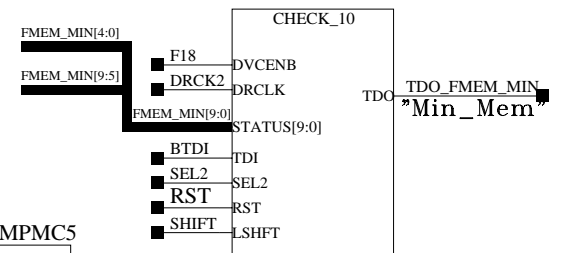
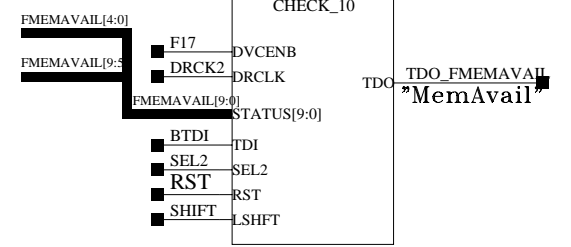
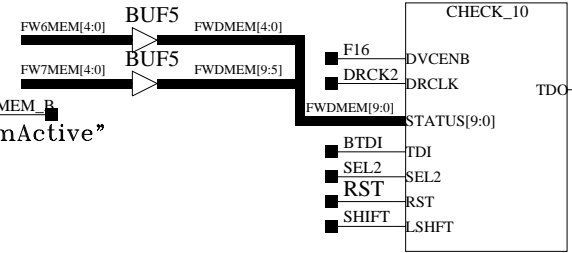
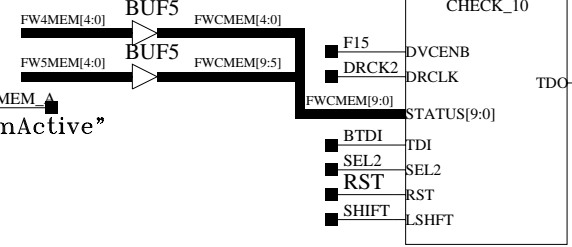
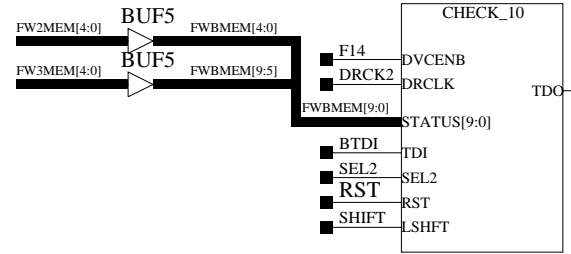
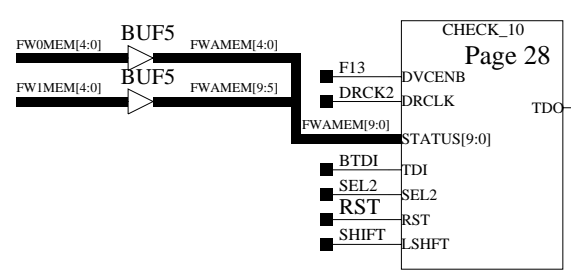
		JRG
Title:		VIRTEX Family Fast M32ESbus Macro
Comments:		32-to-1 bussed Multiplexer with Enable
Date:	29th September 2003	Ver: 1
Sheet Size:	B	Rev: A

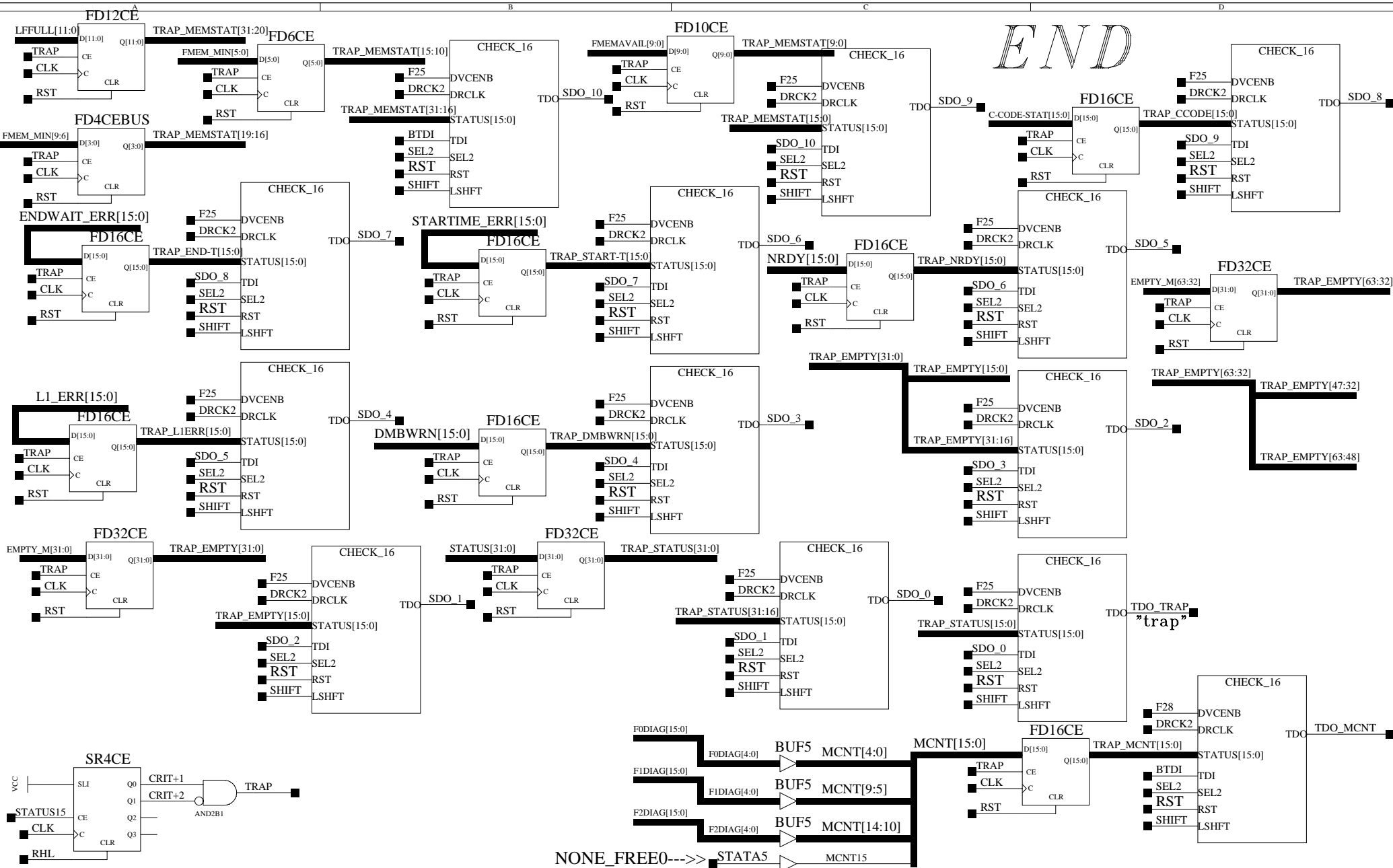
# JTAG Instruction Decode

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]
25	Diagnostic Trap [192-bits]
26	RdCtrl1 Current DDU L1A Number (24-bit scaler)
*28	# Mem assigned to Fibers 2-0 [16-bits]
*30	Fiber 3-0 Mem Diagnostic [InRD0, 4 * 8-bits]
*31	Fiber 7-4 Mem Diagnostic [InRD1, 4 * 8-bits]

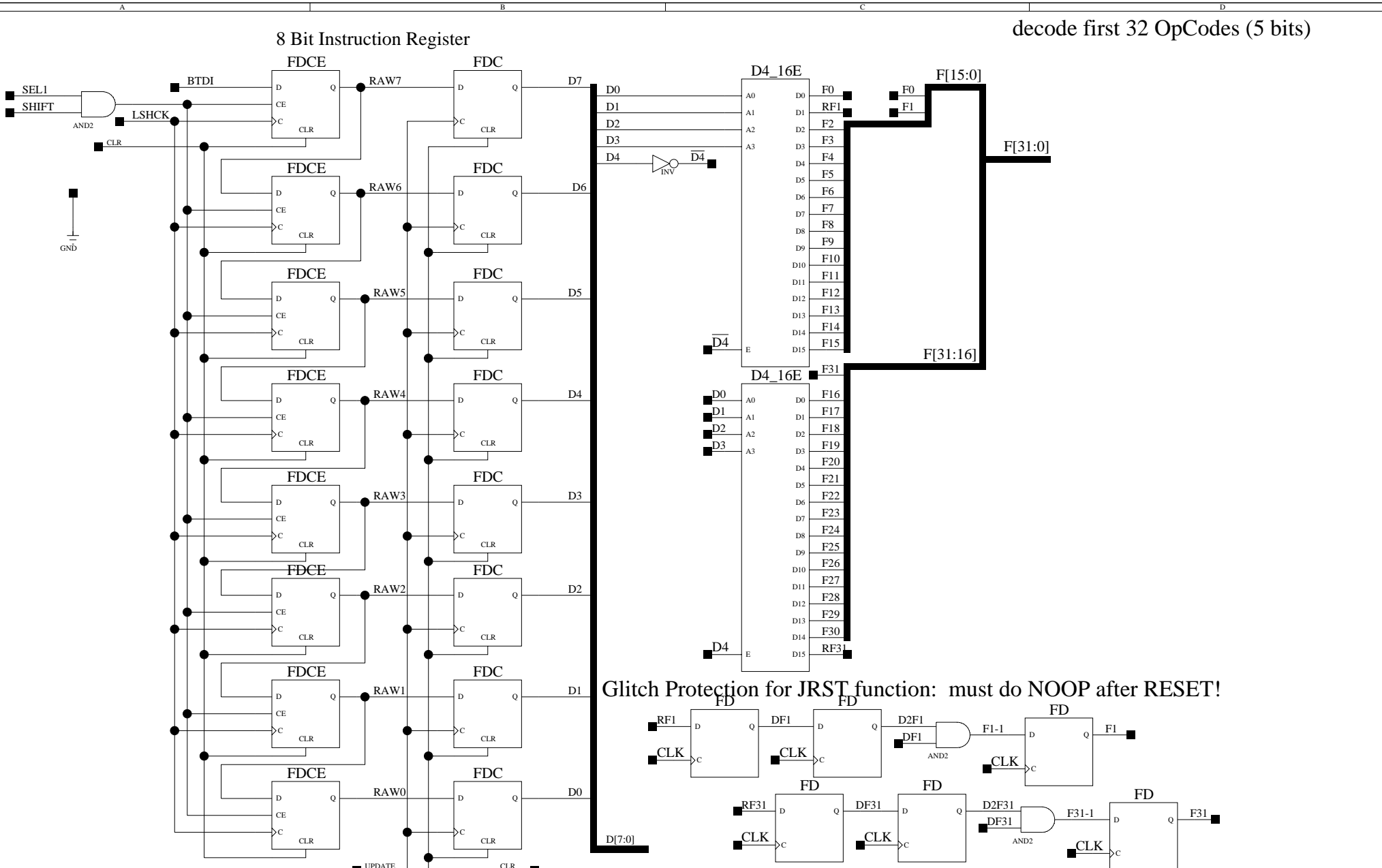




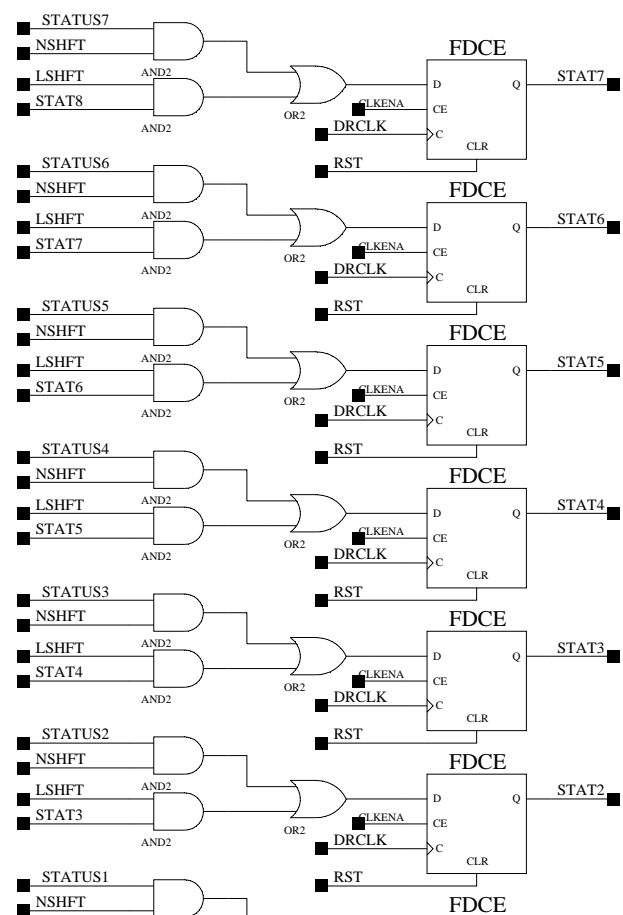
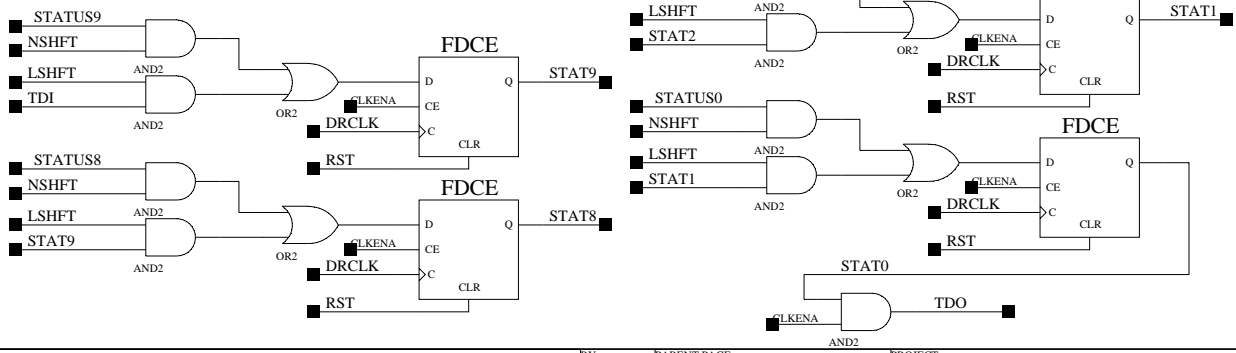
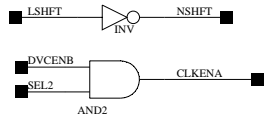
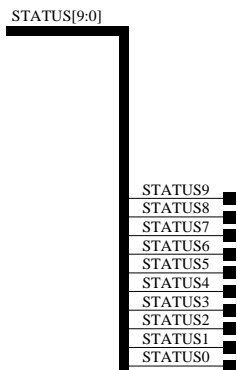




END



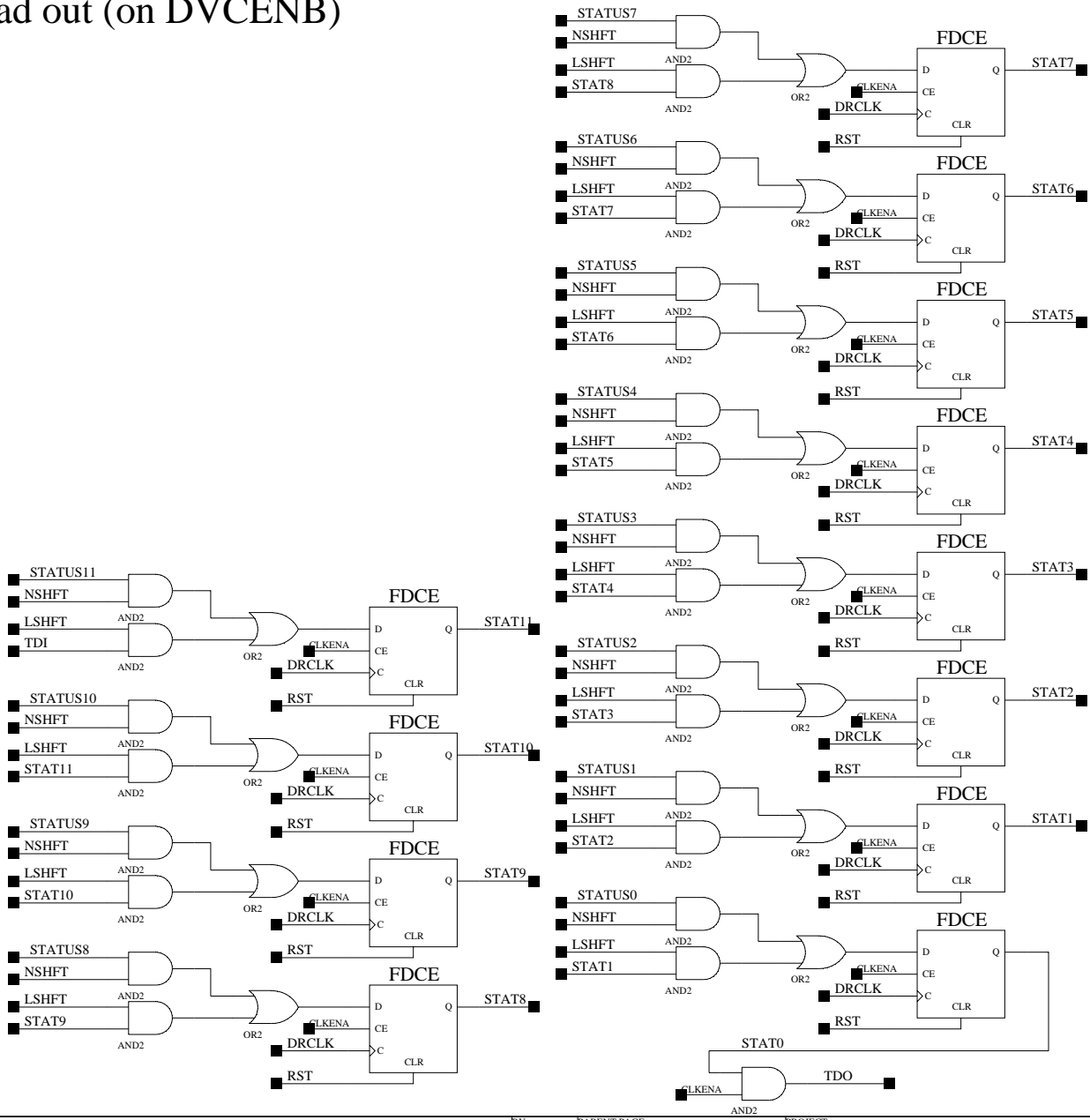
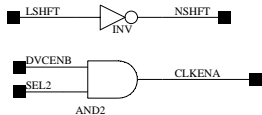
# 10-bit JTAG Register Read out (on DVCENB)



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

STATUS[11:0]

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

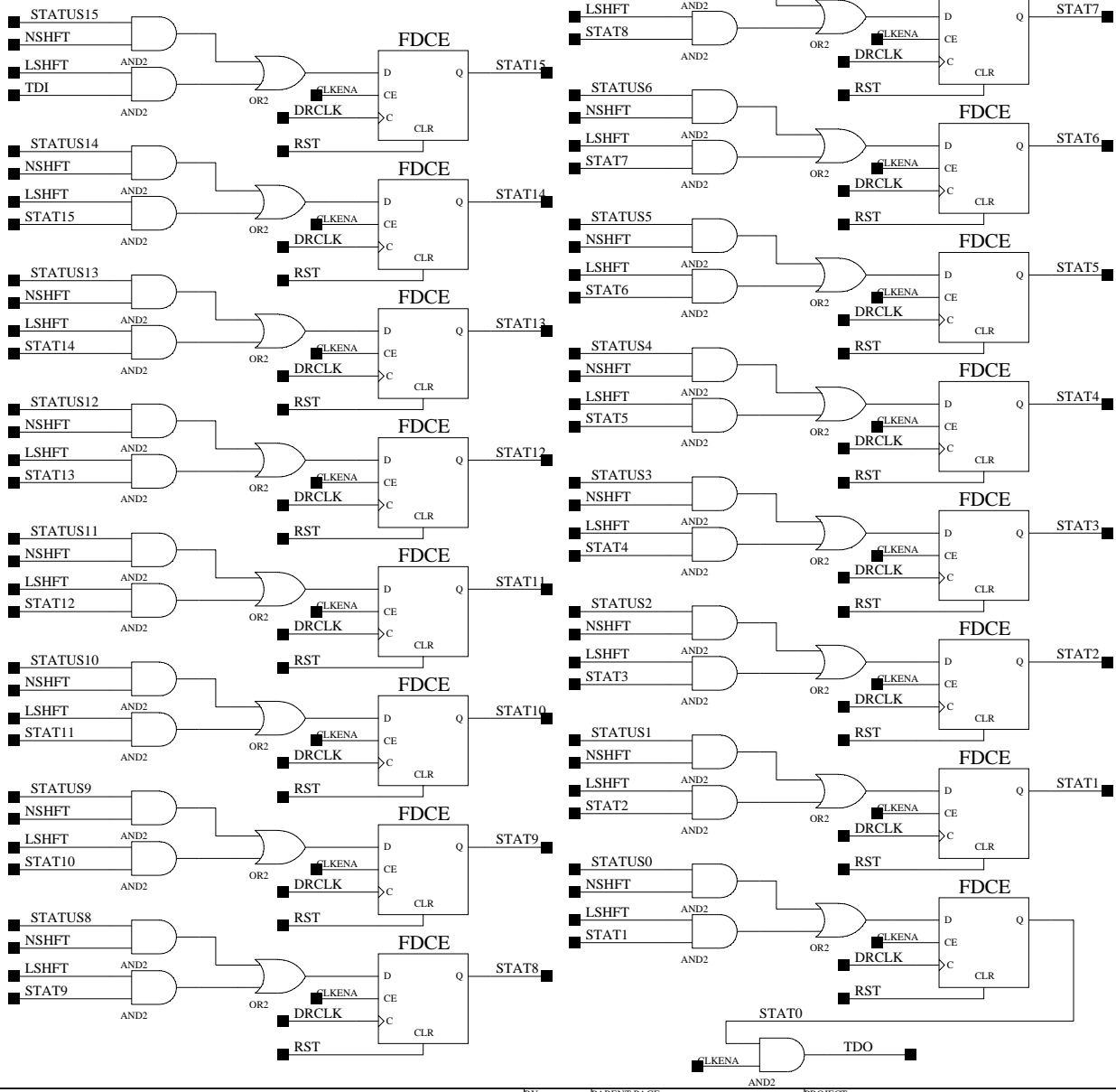
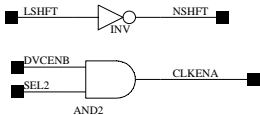




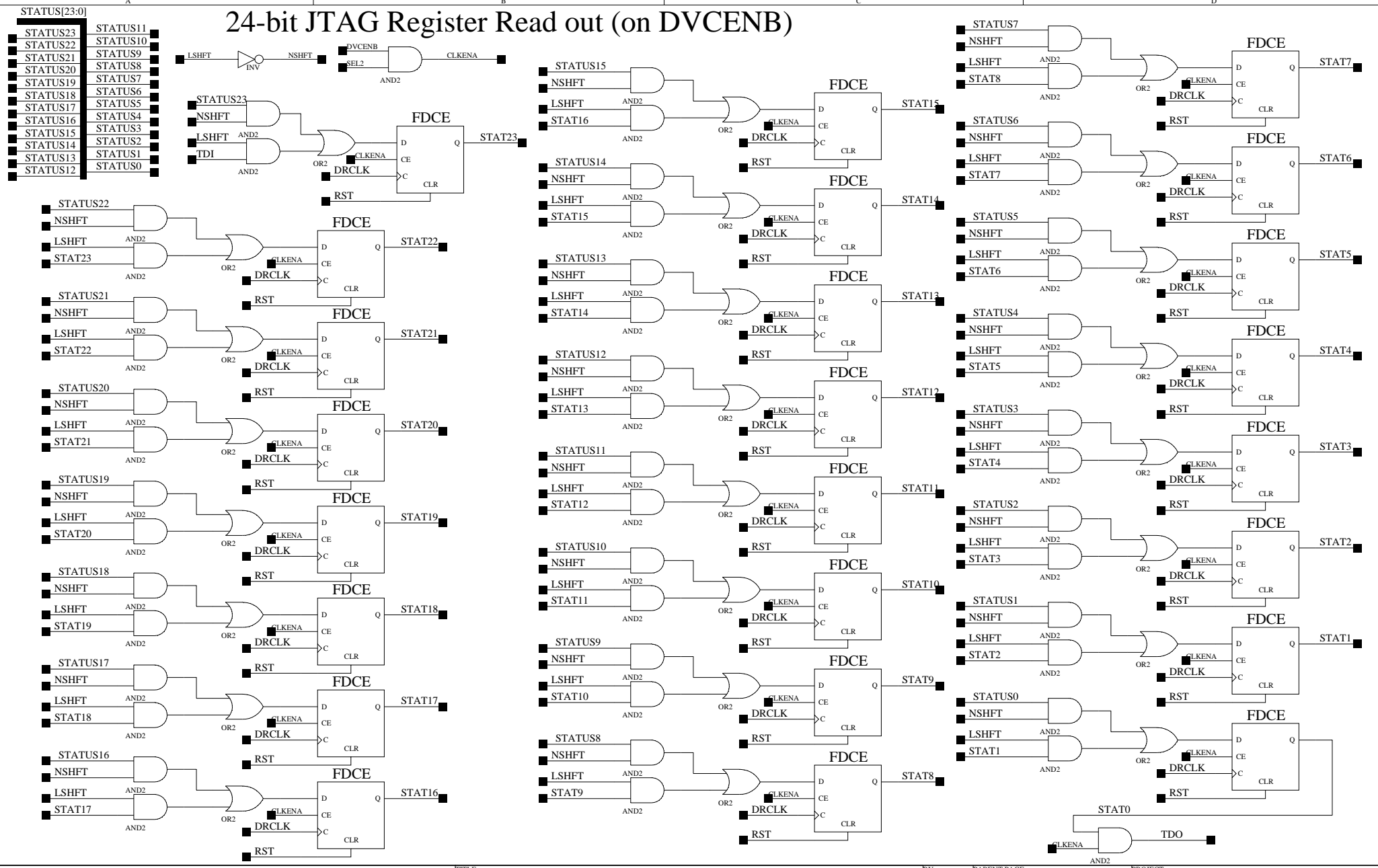
# 16-bit JTAG Register Read out (on DVCENB)

STATUS[15:0]

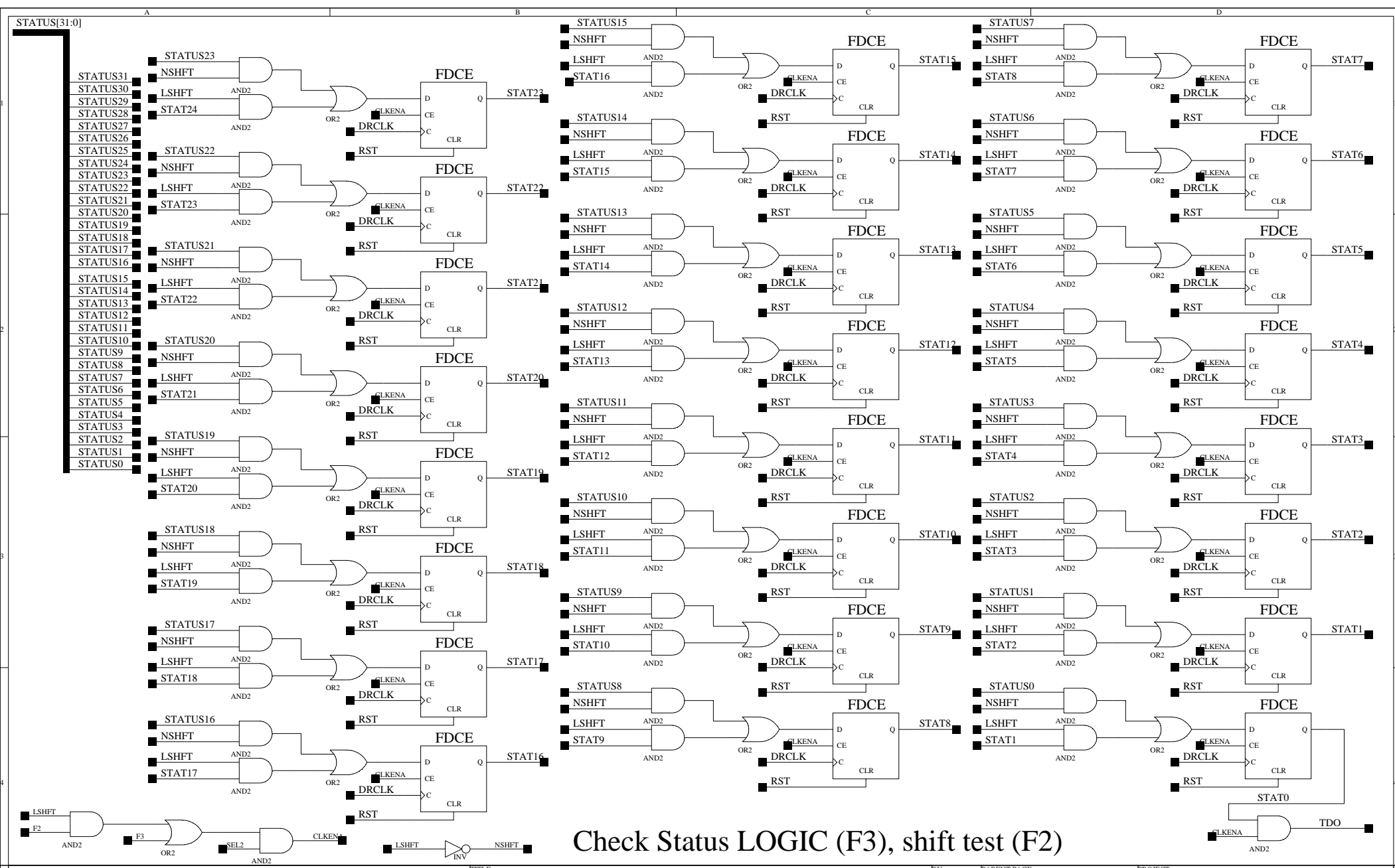
- STATUS15
- STATUS14
- STATUS13
- STATUS12
- STATUS11
- STATUS10
- STATUS9
- STATUS8
- STATUS7
- STATUS6
- STATUS5
- STATUS4
- STATUS3
- STATUS2
- STATUS1
- STATUS0



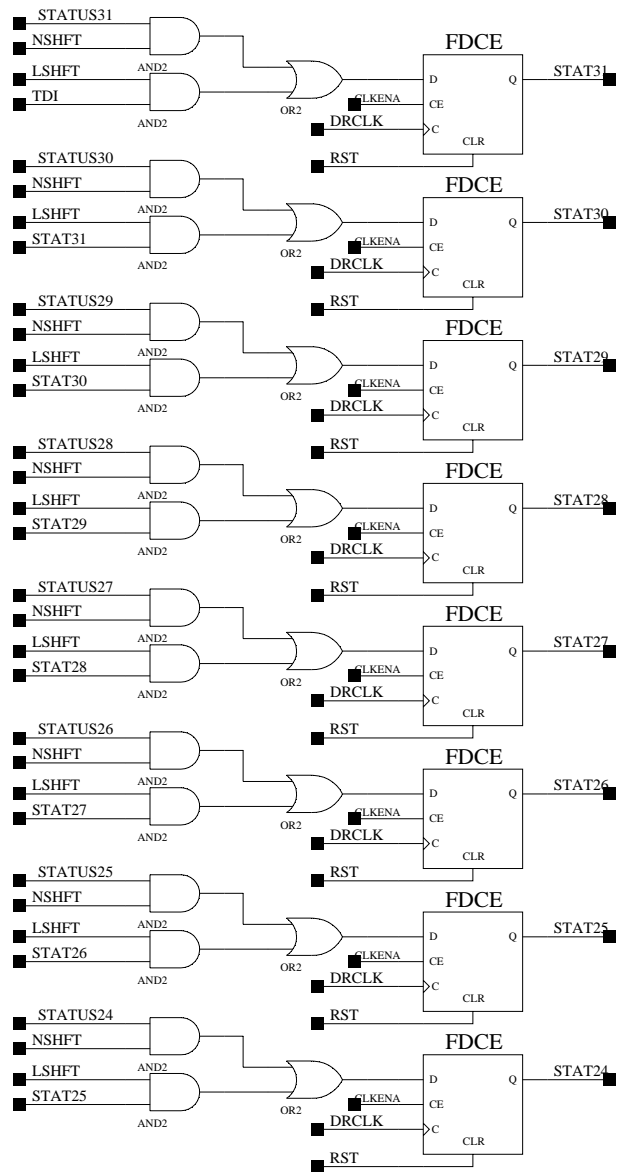
# 24-bit JTAG Register Read out (on DVCENB)



STATUS23:0	STATUS11
STATUS23	STATUS10
STATUS22	STATUS9
STATUS21	STATUS8
STATUS20	STATUS7
STATUS19	STATUS6
STATUS18	STATUS5
STATUS17	STATUS4
STATUS16	STATUS3
STATUS15	STATUS2
STATUS14	STATUS1
STATUS13	STATUS0
STATUS12	



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

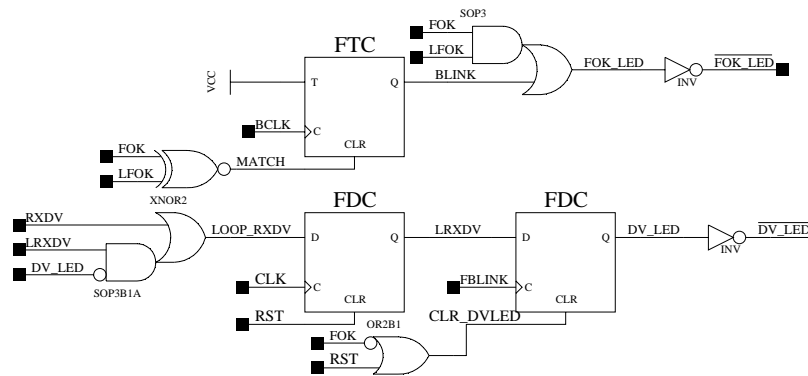


## FOK LED

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

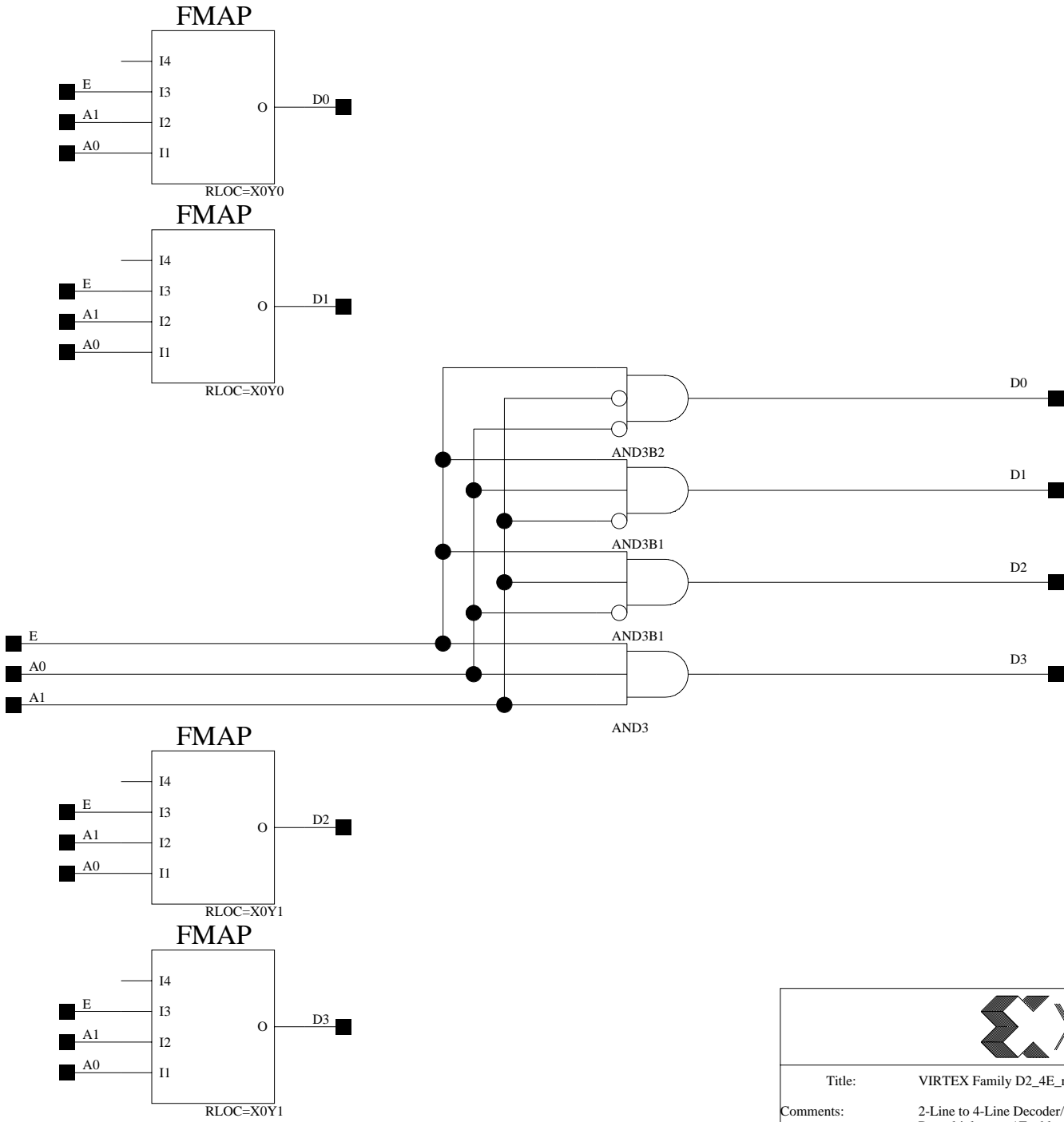
## DAV LED

- 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

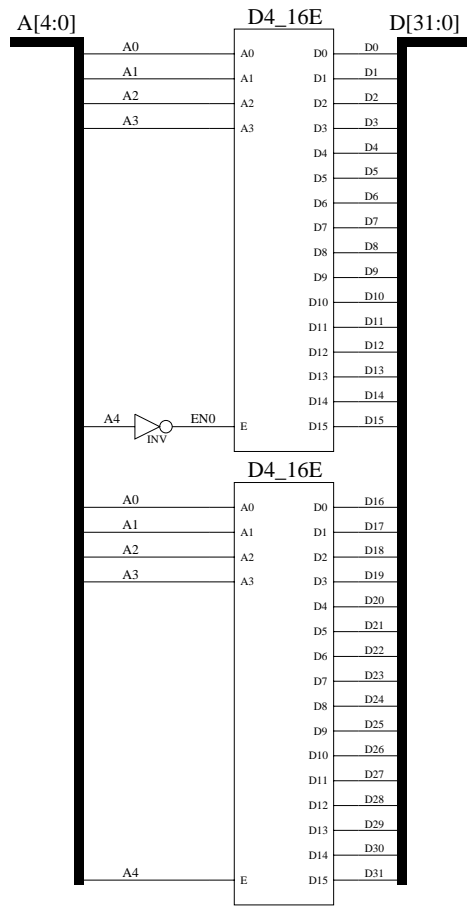


drawn by KS  
 Copyright (c) 1993, Xilinx Inc.



JRG

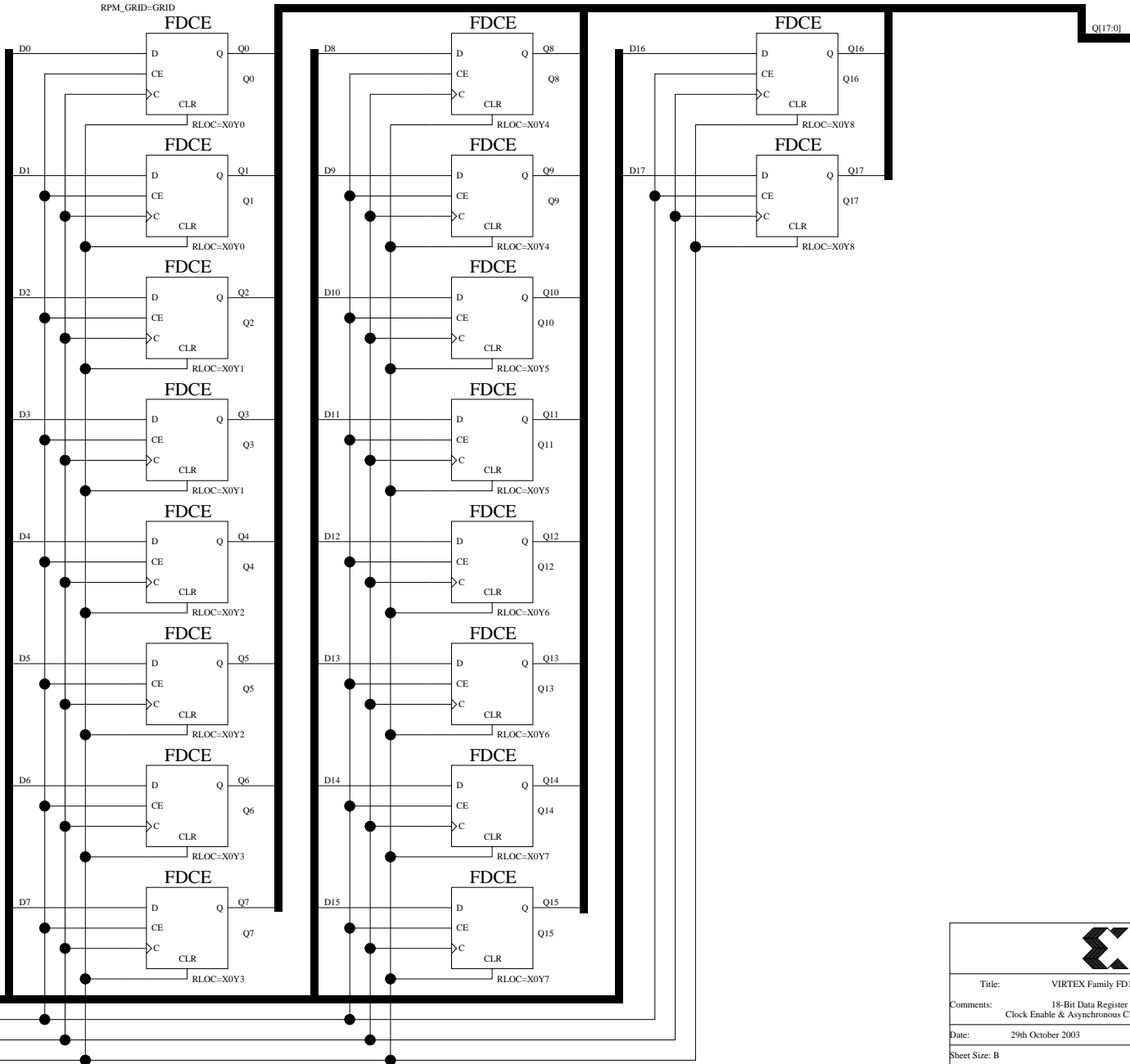
Title:	VIRTEX Family D2_4E_rpm Macro		
Comments:	2-Line to 4-Line Decoder/ Demultiplexer w/ Enable, RPM		
Date:	11th November 2003	Ver:	1
Sheet Size:	A	Rev:	A



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.

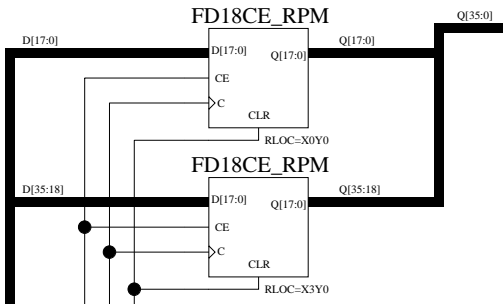
RPM\_GRID=GRID



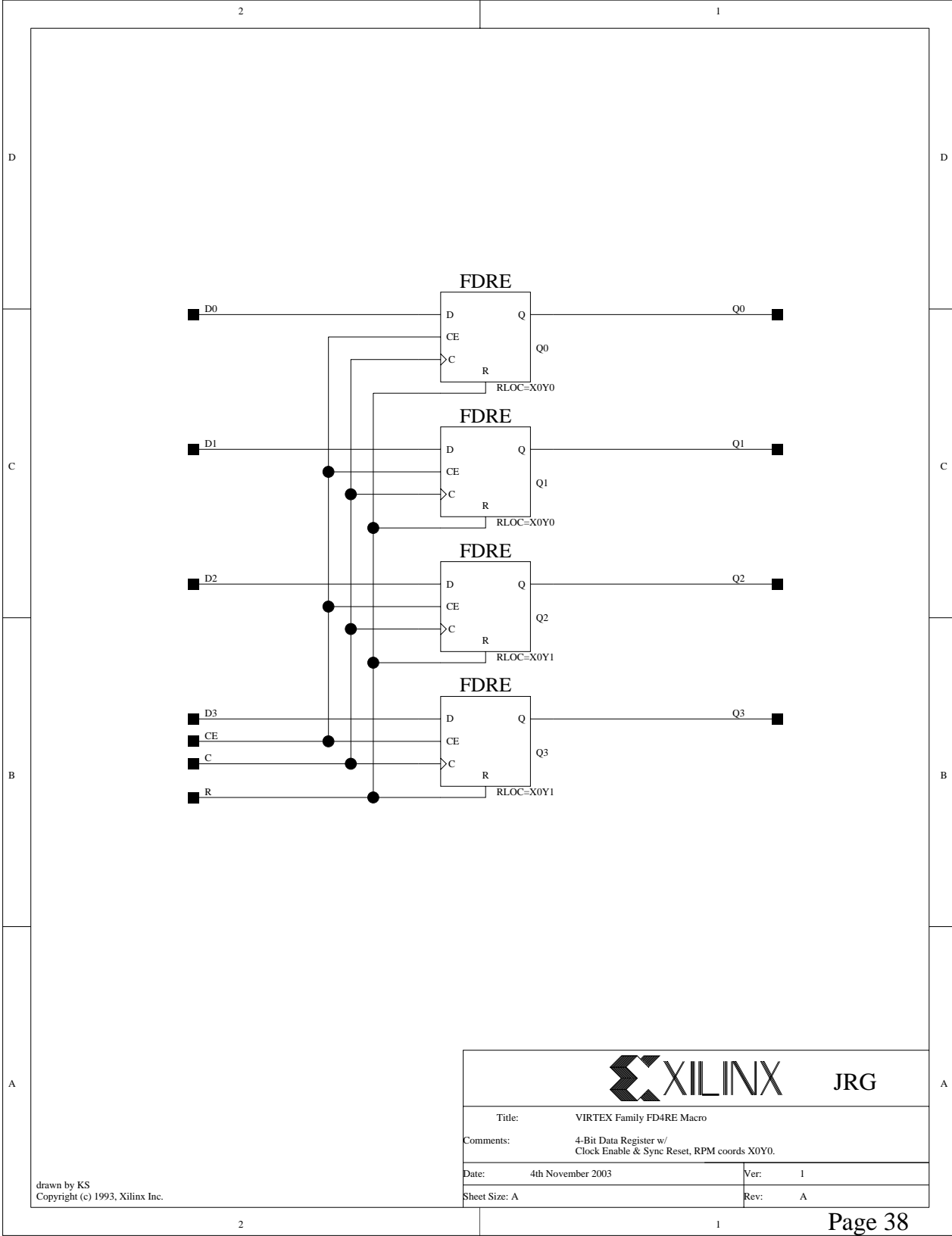
Title:	VIRTEX Family FD18CE_RPM Macro		<b>JRG</b>
Comments:	18-Bit Data Register w/ Clock Enable & Asynchronous Clr, GRID Coords X0Y0-X0Y8		
Date:	29th October 2003	Ver:	1
Sheet Size:	B	Rev:	A



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



Title:	VIRTEX Family FD36CE_RPM Macro	<b>JRG</b>
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

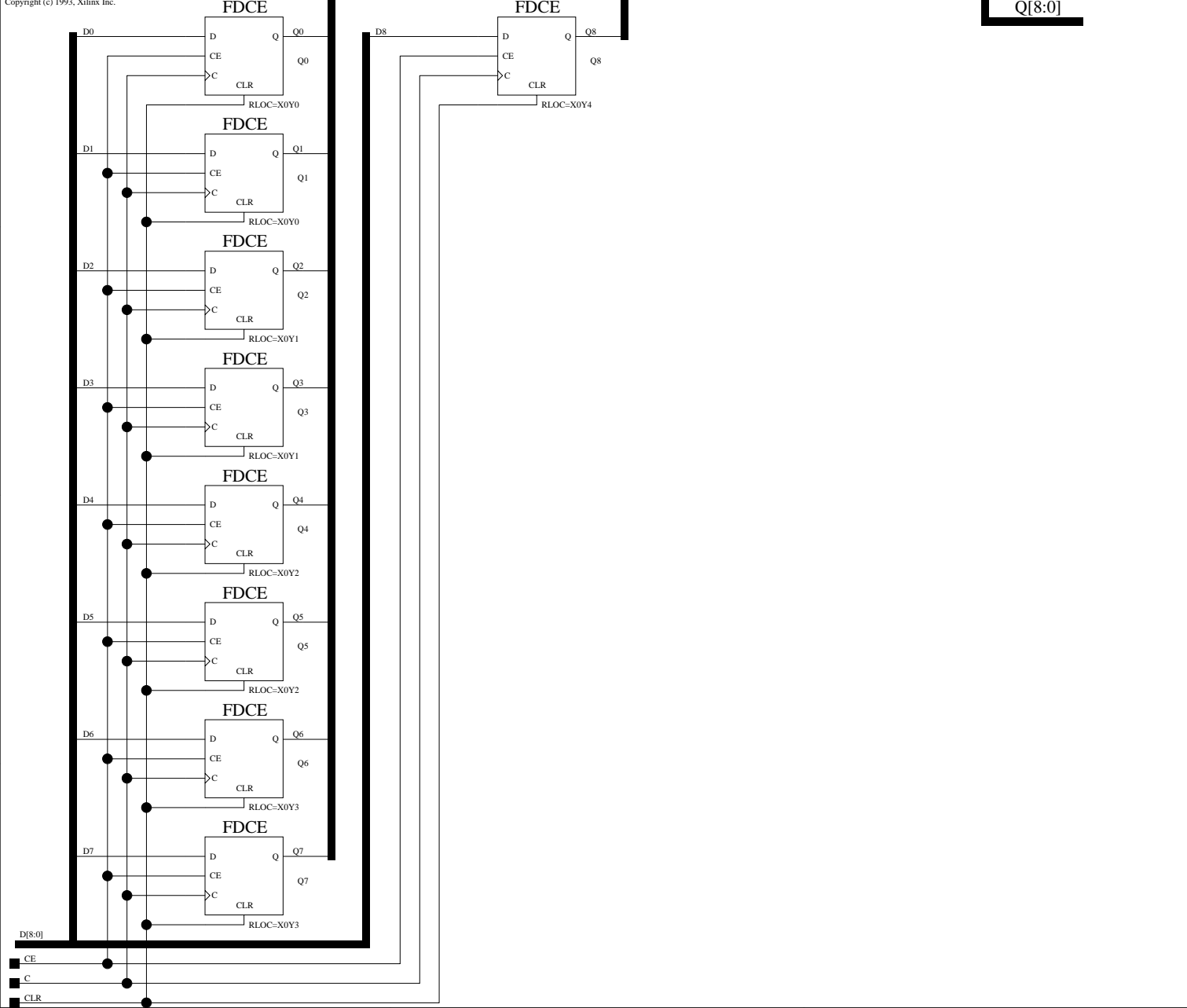



Title:	VIRTEX Family FD4RE Macro		
Comments:	4-Bit Data Register w/ Clock Enable & Sync Reset, RPM coords X0Y0.		
Date:	4th November 2003	Ver:	1
Sheet Size:	A	Rev:	A

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

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

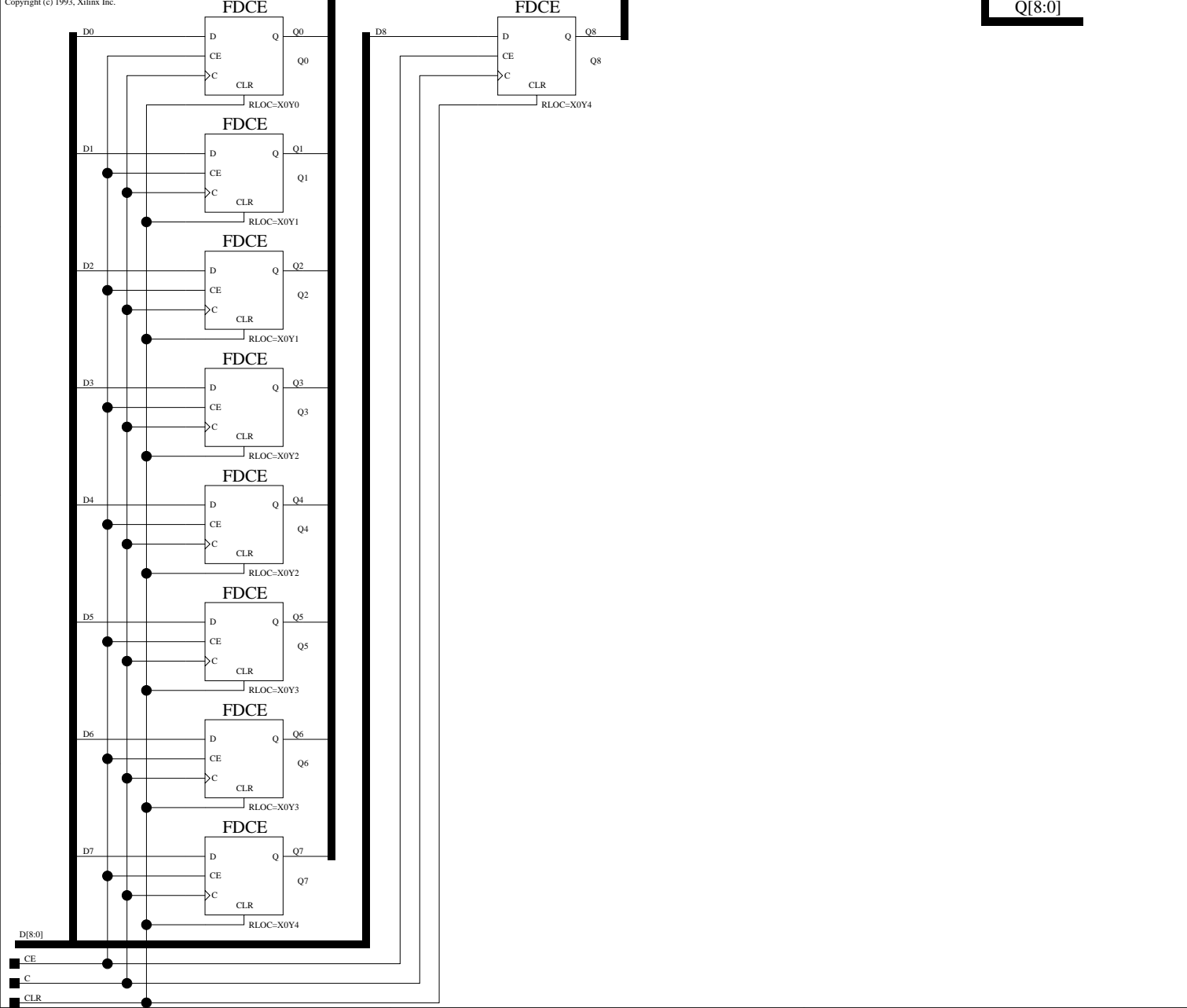
RPM\_GRID=GRID




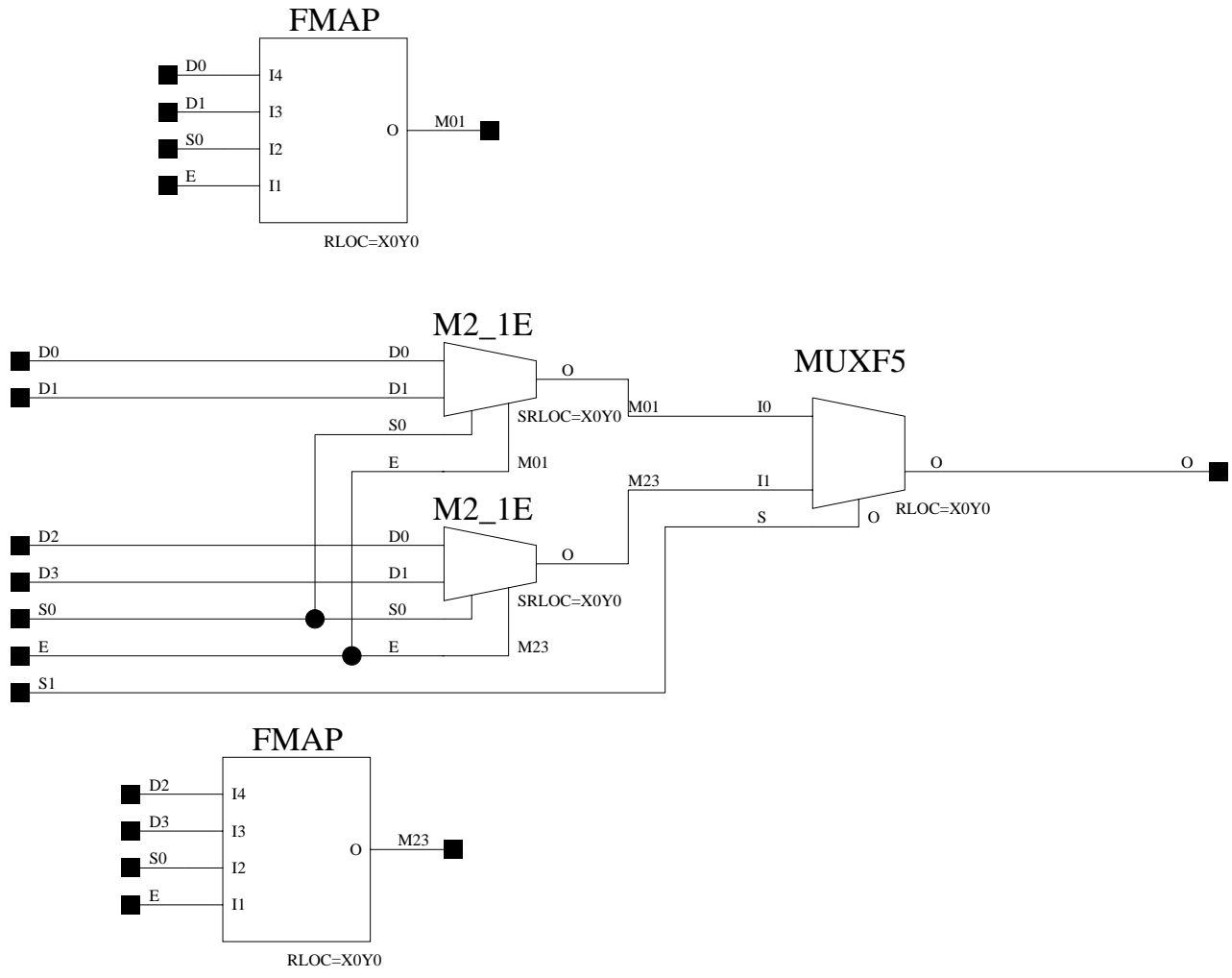
		<b>JRG</b>
Title: VIRTEX Family FD9CE_G Macro		
Comments: 18-Bit Data Register w/ Clock Enable & Asynchronous Clr, RPM GRID		
Date: 10th November 2003	Ver: 1	
Sheet Size: B	Rev: A	

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	

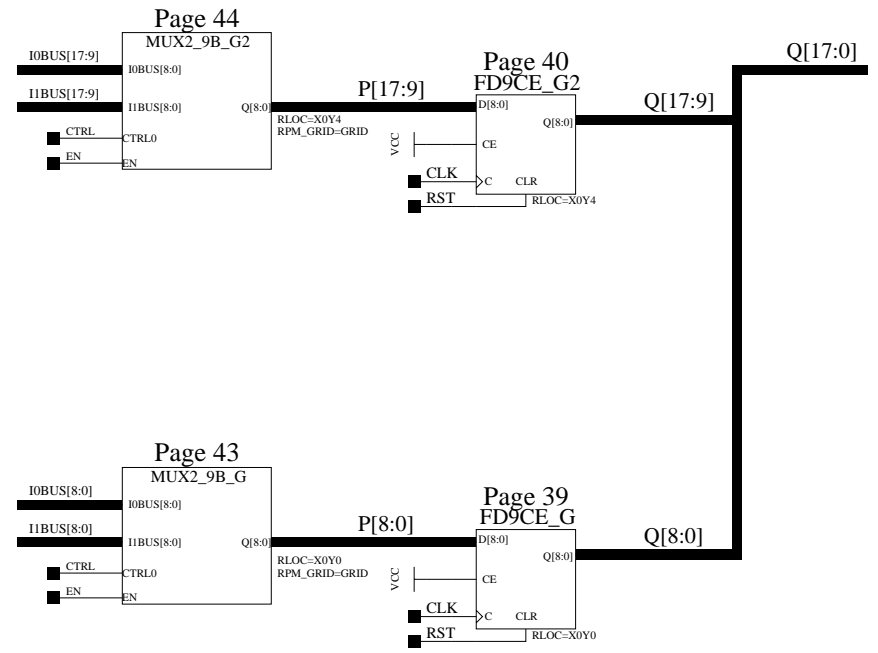


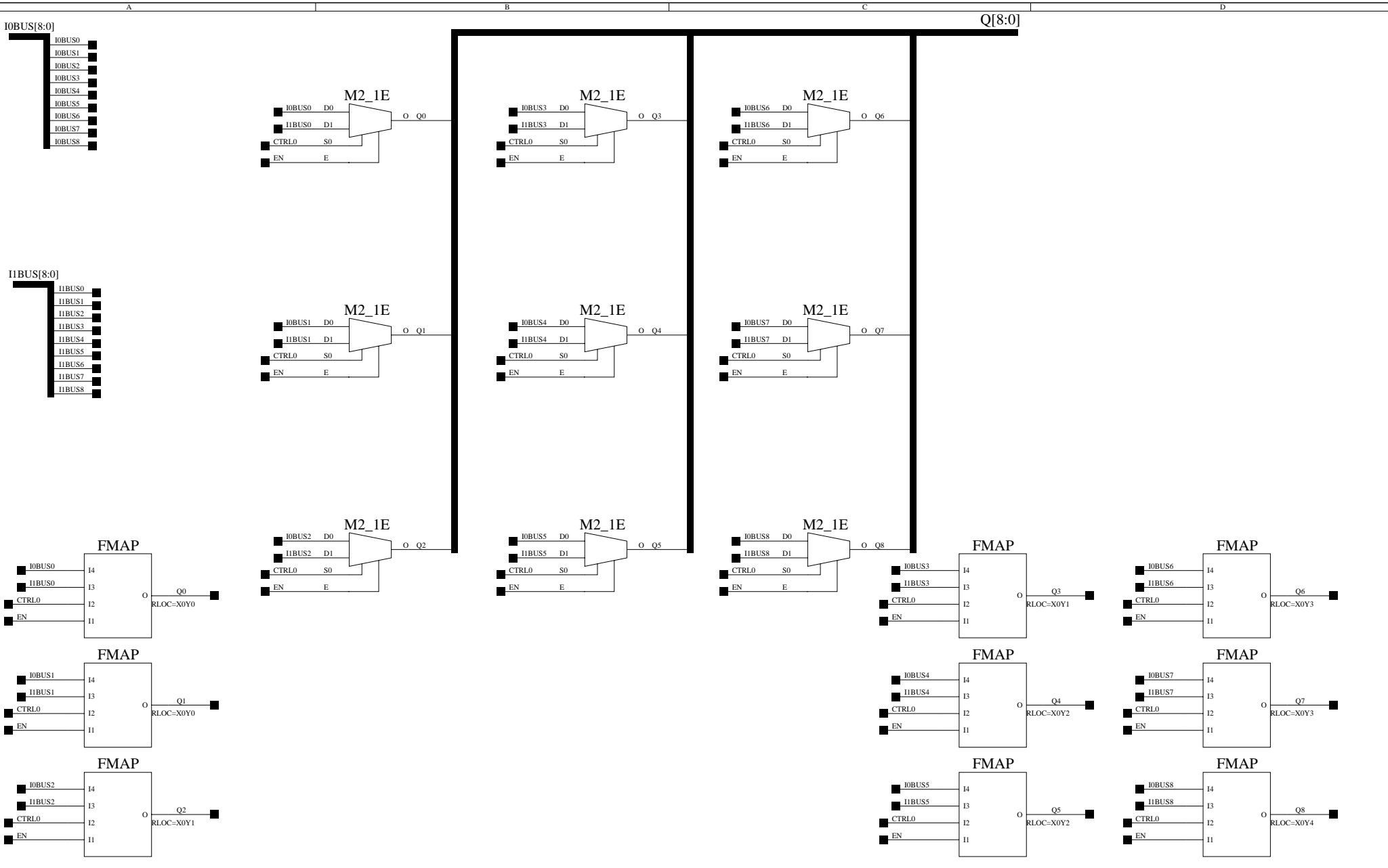
JRG

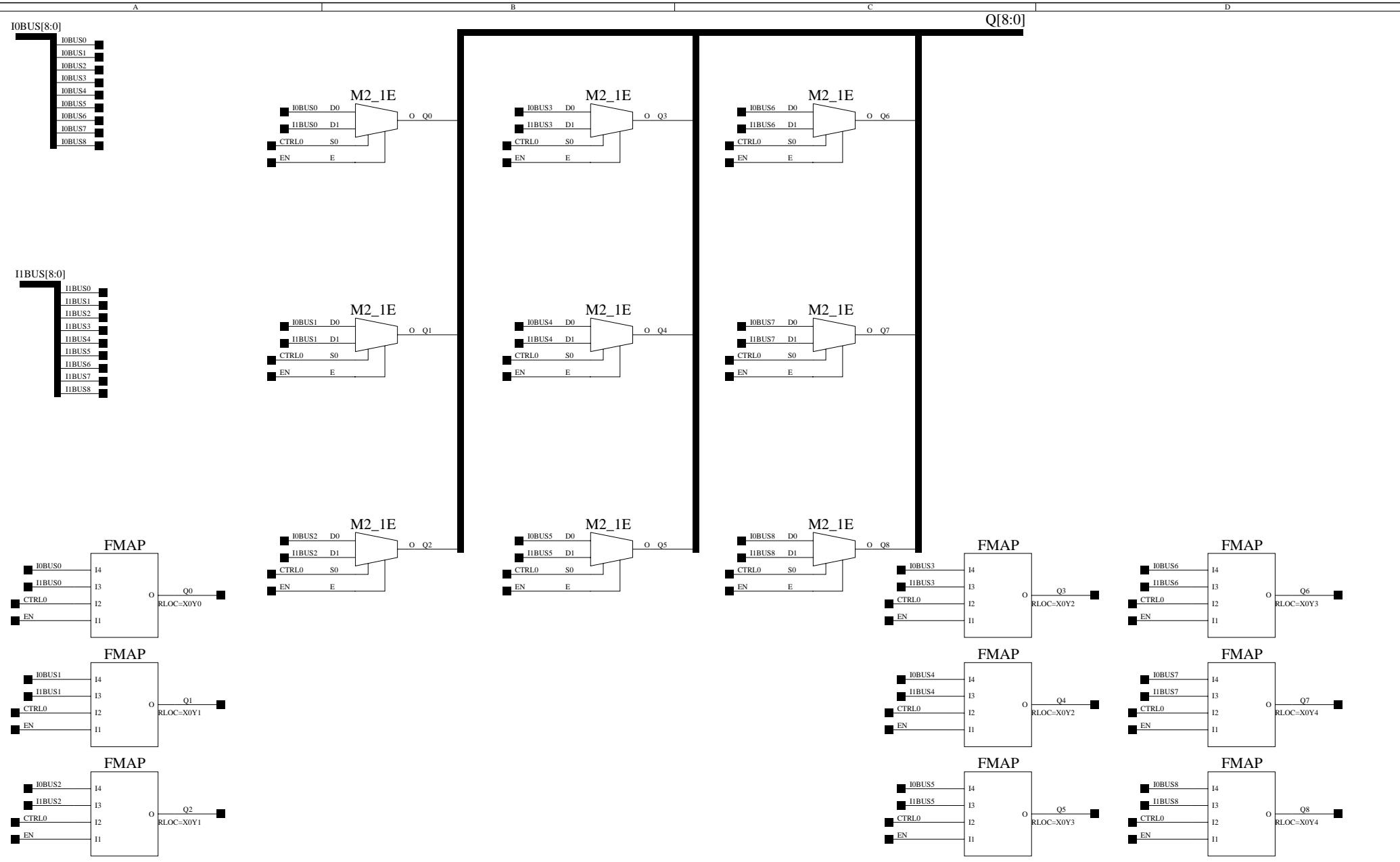
Title:	VIRTEX Family M4_1E_RPM Macro		
Comments:	4-to-1 Multiplexer with Enable		
Date:	31st October 2003	Ver:	1
Sheet Size:	A	Rev:	B

IOBUS[17:0]  
IOBUS[17:9]  
IOBUS[8:0]

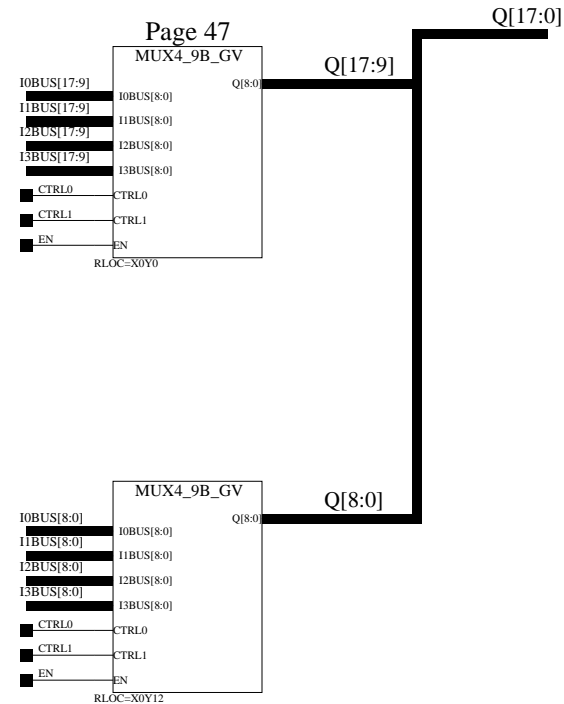
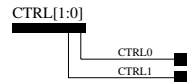
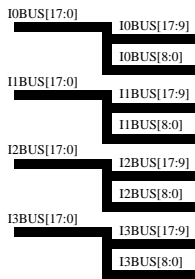
IIBUS[17:0]  
IIBUS[17:9]  
IIBUS[8:0]

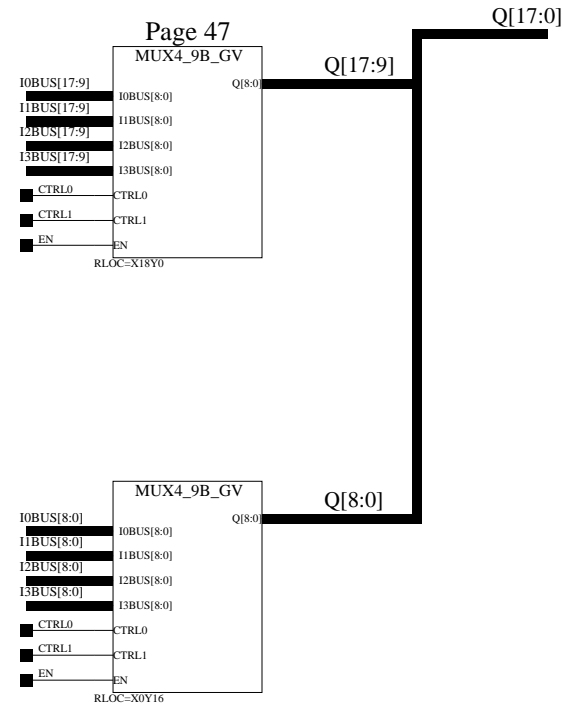
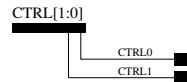
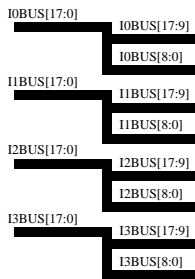


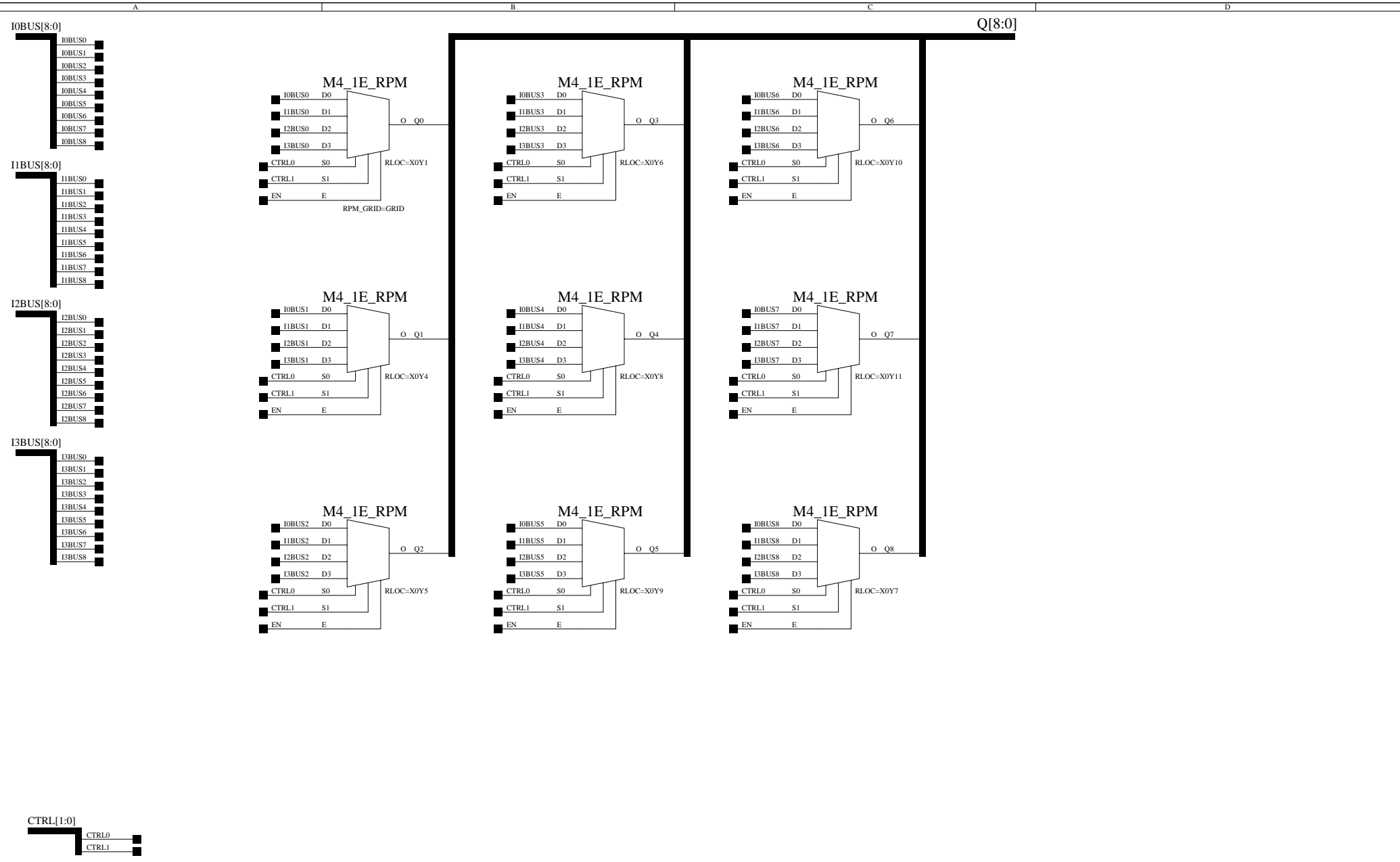




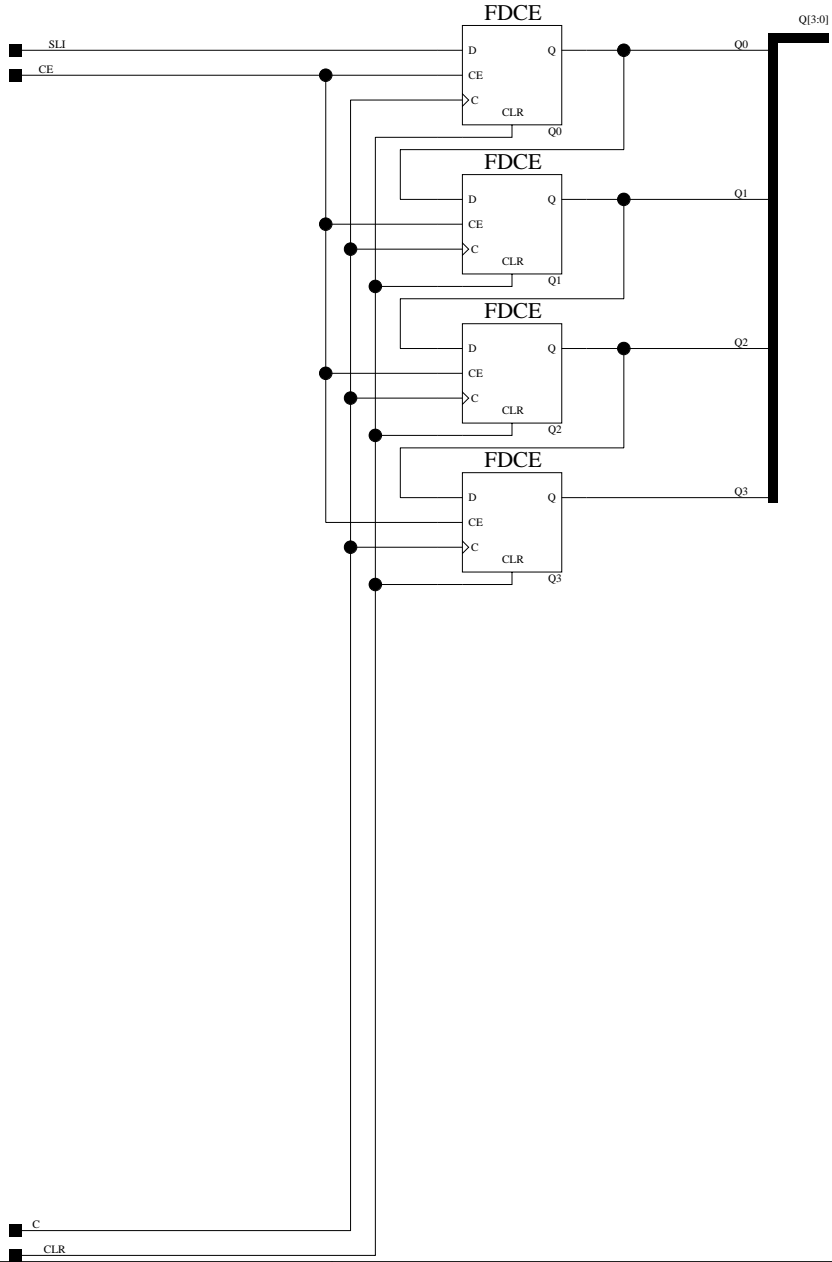








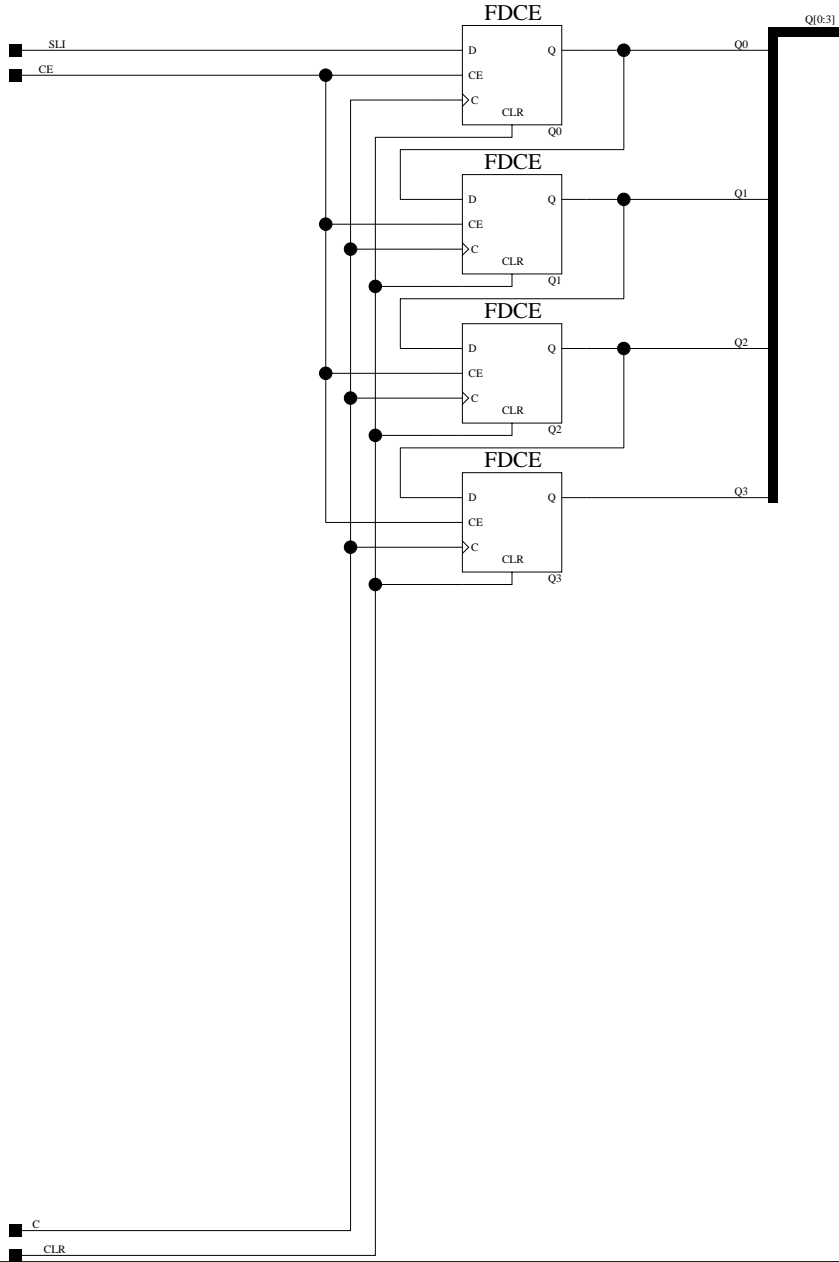
drawn by KS  
Copyright (c) 1993, Xilinx Inc.



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	