A Tour of the XSOC/xr16 Schematics

Version 0.11, February, 2000

Table of Contents

Editor: Jan Gray

1 Introduction

This document provides a brief overview of the XSOC schematics that accompany the Circuit Cellar article series, "Build a RISC System in an FPGA".

XSOC is a simple demonstration system-on-a-chip that includes the xr16 processor core, a memory and on-chip-bus controller, some on-chip RAM, on-chip parallel port input and output, and a bilevel (monochrome) VGA-compatible controller.

XSOC is hosted in the XESS XS40-005XL v1.2 FPGA board, which includes a Xilinx XC4005XL, 12 MHz oscillator, 32 KB SRAM, 8031 MCU, 7-segment LED, voltage regulators, and parallel port and VGA port connectors. We use the RAM for program, data, and video memory.

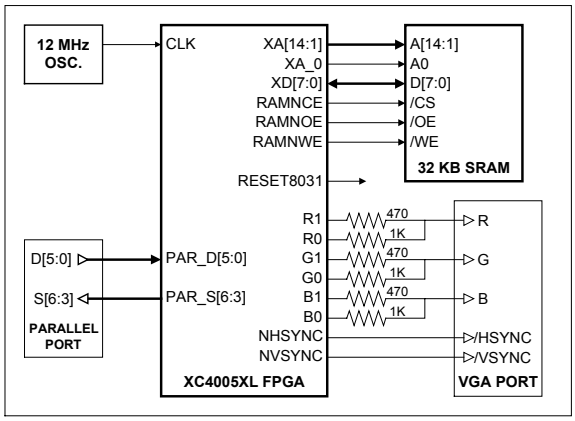


Figure 1: XSOC Interfaces to XS40 Board Resources

Note that $XA_{16:15}$ is connected to $A_{16:15}$ on 128 KB XS40+ boards, with XA[16] strapped to GND.

2 Design Hierarchy

The XSOC schematics are hierarchical. Each block symbol denotes either another schematic (called a *macro* in the Xilinx Foundation documentation), or a primitive.

The XSOC design hierarchy is

XSOC: the top-level system-on-a-chip, including
XR16 *P*, the xr16 RISC processor,
MEMCTRL *MEMCTRL,* the memory controller,
XIN8 *PARIN*, the parallel port input block,
XOUT4 *PAROUT*, the parallel port output block,
XRAM16x16S *IRAM*, on-chip RAM, and
VGA *VGA*, the bilevel (monochrome) display controller.

XR16: the xr16 RISC processor, consisting of
CTRL16 *C*, the control unit, and
DP16: *D*, the datapath.

CTRL16: the xr16 processor control unit, including
FD16CE *NEXTIR*, the next (pending) instruction register,
IRMUX *IRMUX*, the instruction register mux,
FD16CE *IR*, the decode stage instruction register,
FD16CE *EXIR*, the execute stage instruction register,
BUF16 *IRB*, a buffer that cracks IR into the OP, RD, RA, and RB fields,
BUF16 *IMMB,* a buffer that cracks $IR_{11:0}$ into $IMM_{11:0}$,
BUF16 *EXIRB*, a buffer that cracks EXIR into the EXOP, EXRD, and BRDISP fields,
CTRLFSM *FSM*, the control unit state machine,

DECODE *DECODE*, the instruction decoder,
RNMUX4 *RNA*, the port-A register file register selector,
RNMUX4 *RNB*, the port-A register file register selector,
TRUTH *TRUE*, the branch condition taken/not taken evaluator,
FD4PE *T1*, *T2*, the result multiplexer output enable registers

CTRLFSM: the xr16 processor control unit state machines, including these flip-flops:
IF and *DMA*, states in the current memory cycle state machine,
DCANNUL and *EXANNUL*, to annul the decode and/or execute state instructions,
DMAP and *ZEROP*, DMA pending and zero-DMA pending,
INTP and *DCINT*, interrupt pending, and "current decode stage instruction is `int`",
RESET, reset processor on power-on.

DECODE: the xr16 processor instruction decoder, with logic and 16x1-bit ROMs to decode
RRRI, instruction is `rr` or `ri` format,
RI, instruction is `ri` format,
FNSBC, instruction's `fn` field is `sbc`/`sbci`,
FNADCSBC, instruction's `fn` field is `adc`/`adci`/`sbc`/`sbci`,
IMM12, instruction is `imm12` format, e.g. `call`/`imm`,
IMM4, instruction is `rri` or `ri` format and has a 4-bit immediate,
SEXTIMM4, instruction's 4-bit immediate get sign-extended,
WORDIMM4, instruction's 4-bit immediate encodes a word displacement,
FNSRA, instruction's `fn` field is `sra`,
EXFNSRA, execute stage instruction's `fn` field is `sra`,
NSUB, instruction is *not* a subtract,
EXNSUB, execute stage instruction is *not* a subtract,
ADDSUB, instruction is an `add`/`sub`/`addi`,
SUB, instruction is `sub`,
NLW, instruction is not `lw`,
NLD, instruction is not a load, e.g. not `lw`/`lb`,
NLB, instruction is not `lb`,
NJAL, instruction is not a jump, e.g. not `jal`/`call`,
EXIMM, execute stage instruction is `imm` prefix,
EXLDST, execute stage instruction is a load or store,
EXLBSB, execute stage instruction is `lb`/`sb`,
EXRESULTS, execute stage instruction writes a result to the register file,
EXJAL, execute stage instruction is a jump, e.g. `jal`/`call`,
JALI, instruction is `call`,
EXJALI, execute stage instruction is `call`,
ST, instruction is a store, e.g. `sw`/`sb`,
EXST, execute stage instruction is a store, e.g. `sw`/`sb`,
BR, instruction is a branch,
ADCSBC, instruction is `adc`/`adci`/`sbc`/`sbci`,
NSUM, not (instruction result is the adder/subtractor)
NLOGIC, not (instruction result is the logic unit),
NSR, not (instruction result is the right shifter),
NSL, not (instruction result is the left shifter),
DCINTINH, instruction is interlocked and therefore inhibits a pending interrupt.

RNMUX4: the 4-bit register selector multiplexer, floorplanned 4Rx1C, consisting of
RNMUX1 *R0*, *R1*, *R2*, *R3,* 1-bit register selector muxes, and
FWD *FWD1*, the result forwarding detector.

RNMUX1: the 1-bit register selector multiplexer, floorplanned 1Rx1C, including
M2_1H *RN0*, the source register index/destination register index mux.

FWD: the result forwarding logic; the result should be forwarded to the operand register when the decode stage source register equals the execute stage destination register, unless the destination register is `r0`.

TRUTH: the branch condition taken/not taken evaluator, floorplanned 3Rx2C.

IRMUX: a 16-bit 2-1 mux with special *INT* input to force the output to be the `int` instruction (`jal r14,10(r0)`), e.g. 0xAE01.

DP16: the xr16 processor datapath, floorplanned 9Rx10C, including
REGFILE *AREGS*, the port A copy of the register file,
REGFILE *BREGS*, the port B copy of the register file,
M2_16 *FWD*, the A operand result forwarding multiplexer,
IMM16 *IMMED*, the B operand immediate operand multiplexer,
FD16E *A*, the A operand,
FD12E4E *B*, the B operand,
FD16E *DOUT*, the store data output register,
ADSU16 *ADDSUB*, the ALU adder/subtractor,
LOGIC16 *LOGIC*, the ALU logic unit,
ZERODET *Z*, the zero result detector,
PCDISP16 *PCDISP*, the PC displacement multiplexer,
ADD16 *PCINCR*, the PC incrementer,
M2_16Z *ADDRMUX*, the next-address multiplexer,
RAM16X16S *PC*, the program counter register file,
FD16E *RET*, the return address register,
BUFT16X *SUMBUF, LOGICBUF, SRBUF, SLBUF, RETBUF*, the result multiplexer,
BUF8X *LDBUF, UDBUF, UDLDBUF,* the store data output drivers, and
BUF8X *ZHBUF*, the upper-byte zero-extension result driver.

REGFILE: a single ported 16x16 register file, floorplanned 8Rx1C, consisting of
REGFILE2 *R0, R2, R4, R6, R8, R10, R12, R14*.

REGFILE2: a 16x2-bit slice of the register file, floorplanned 1Rx1C.

IMM16: a complex 16-bit multiplexer of the 16-bit register operand and various immediate values formed from the 4- and 12-bit immediate operand fields. It is floorplanned 8Rx1C. It includes
M2_16K *M0*, the underlying 16-bit 2-1 mux.

LOGIC16: a 16-bit logic unit computing $A \wedge B$, $A \vee B$, $A \oplus B$, or $A \wedge \neg B$, floorplanned 8Rx1C, consisting of
LOGIC4 *L0, L4, L8, L12*, 4 4-bit logic units.

LOGIC4: a 4-bit slice of the logic unit, floorplanned 2Rx1C, consisting of
LOGIC1 *L0, L1, L2, L3*, 4 1-bit logic units.

LOGIC1: a 1-bit slice of the logic unit, floorplanned ½Rx1C.

ZERODET: a 16-bit zero detector, floorplanned (sparsely, 4 FMAPs total) 7Rx1C.

PCDISP16: a 16-bit PC displacement multiplexer, selecting +2 or sign-ext($BRDISP_{7:0}||0$), floorplanned 5Rx1C, including
M2_4 *M0, M4*, and
M2_1F *M8*, multiplexers.

RAM16X16S: a 16x16-bit synchronous SRAM, floorplanned 8Rx1C, consisting of
RAM16X8SX *R0, R8*, lower and upper bytes of RAM.

RAM16X8SX: a 16x8-bit synchronous SRAM, floorplanned 4Rx1C.

BUFT16X: a 16-bit 3-state buffer with active low output enable T, floorplanned 8Rx1C, consisting of
BUFT8X *B0, B8*, lower and upper byte buffers.

BUFT8X: an 8-bit 3-state buffer with active low output enable T, floorplanned 4Rx1C.

MEMCTRL: external memory and on-chip bus controller, including these flip-flops and outputs:
IO, current access is to on-chip bus memory mapped I/O,
RAMRD, current access is a read from external SRAM,
RAMWR, current access is a write to external SRAM,
WORD, current access is a 16-bit word,
$A_{4:0}$, current address LSBs,
$SEL_{7:0}$, next address selects on-chip peripheral$_i$,
CTRL, the abstract on-chip bus control bus, including $CTRL_0$, I/O not ready pull-down input,
W12, memory write half cycles W1, W2,
W34, memory write half cycles W3, W4,
W56, memory write word half cycles W5, W6,
W23_45, memory write half cycles W2, W3, and (if word write) W4, W5,
W45, memory write word half cycles W4, W5,
UDTN, *LDTN*, upper and lower byte *next* access I/O read output enables,
UCEN, LCEN, upper and lower byte *next* access I/O write clock enables,
UXDT, LXDT, upper and lower byte memory read external RAM output enables,
XDOUTT, memory write external data output enables,
LDT, processor store data lower byte output enable,
UDT, processor store data upper byte output enable,
UDLDT, processor store data upper byte to lower byte output enable,
RAMNCE, external RAM active low clock enable,
RAMNOE, external RAM active low output enable,
RAMNWE, external RAM active low write enable.

XIN8: 8-bit input, consisting of
DCTRL *D*, the peripheral's control bus decoder.

DCTRL: abstract control bus decoder, including these flip-flops
UCE, LCE, this peripheral's upper and lower byte *current* access I/O write clock enables,
UDT, LDT, this peripheral's upper and lower byte *current* access I/O read output enables.

XOUT4: 4-bit output register, consisting of
DCTRL *D*, the peripheral's control bus decoder,
FD4RE *Q,* the output register.

XRAM16X16: 16x16-bit on-chip SRAM, floorplanned 8Rx1C, consisting of
DCTRL *D*, the peripheral's control bus decoder,
RAM16X8SX *RU, RL,* upper and lower byte 16x8 RAMs,
BUFT8X *UD, LD*, upper and lower byte output enables.

VGA: bilevel (monochrome) VGA timing compatible video controller, consisting of
LSFR10 *H*, horizontal double-pixel counter,
LFSR10 *V*, vertical line counter,
EQ10 *H0, HBLANK, HSYNCON, HSYNCOFF*, horizontal counter comparators,
EQ10 *V0, VBLANK, VSYNCON, VSYNCOFF,* vertical counter comparators,
FJKCE (J-K flip flop) *HENN*, horizontal counter pixel enable on *next* cycle,
FDCE *HEN*, horizontal counter pixel enable,
FJKCE *NHSYNC,* not horizontal sync,
FJKCE (J-K flip flop) *VEN*, vertical counter pixel enable
FJKCE *NVSYNC,* not vertical sync,
CB4RE *DOTCNT*, double-pixel counter,
WORD, signal to fetch new 16-bit video data word,
FD16E *PIXELS*, pending pixels buffer register,

M2_16 *PMUX*, current pixels shifter,
FD16E *P*, current pixels register,
M2_1E *PIXEL*, current double-pixel to pixel multiplexor and output,
RESET, signal to reset external DMA counter in anticipation of next video frame.

LFSR10: 10-bit linear feedback shift register counter with force-cycle input and clock enable.

EQ10: 10-bit equality comparator.

GENERAL PURPOSE MULTIPLEXERS

M2_16Z: 16-bit 2-1 multiplexer, with *force zero* control, floorplanned 8Rx1C, consisting of
M2_16K *M0*.

M2_16K: 16-bit 2-1 multiplexer, with *force constant* control, floorplanned 8Rx1C, consisting of
M2_4K *M0, M4, M8, M12*.

M2_4K: 4-bit 2-1 multiplexer, with *force constant* control, floorplanned 2Rx1C, consisting of
M2_1K *M0, M1, M2, M3*.

M2_1K: 1-bit 2-1 multiplexer, with *force constant* control, floorplanned ½Rx1C.

M2_16: 16-bit 2-1 multiplexer, floorplanned 8Rx1C, consisting of
M2_16K *M0*.

M2_4: 4-bit 2-1 multiplexer, floorplanned 2Rx1C, consisting of
M2_1F *M0, M1, M2, M3*.

M2_1F: 1-bit 2-1 multiplexer, floorplanned ½Rx1C.

M2_1H: 1-bit 2-1 multiplexer, floorplanned 1Rx1C (HMAP).

GENERAL PURPOSE REGISTERS (FLIP-FLOP VECTORS)

FD16E: 16-bit register with clock enable, floorplanned 8Rx1C, consisting of
FD4E *Q0, Q4, Q8, Q12*.

FD4E: 4-bit register with clock enable, floorplanned 2Rx1C.

FD12E4E: 16-bit register with separate clock enables for $Q_{15:4}$ and $Q_{3:0}$, consisting of
FD4E *Q0, Q4, Q8, Q12*.

FD4PE: 4-bit register with clock enable, floorplanned 2Rx1C, preset (resets to `4'b1111`).

STANDARD LIBRARY MACROS used in this design include
ADSU16, 16-bit adder/subtractor,
ADD16, 16-bit adder,
D3_8E, 3-to-8 decoder with enable.

3 Floorplan

Here is the floorplan of the XSOC design in an XC4005XL-PC84C.

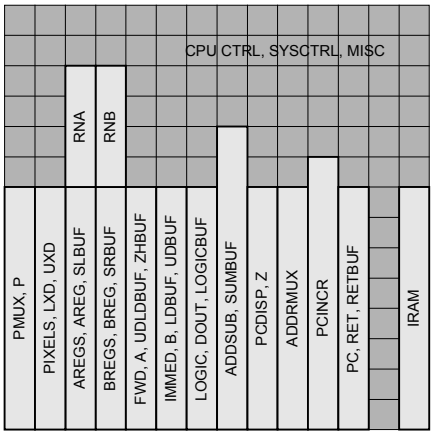
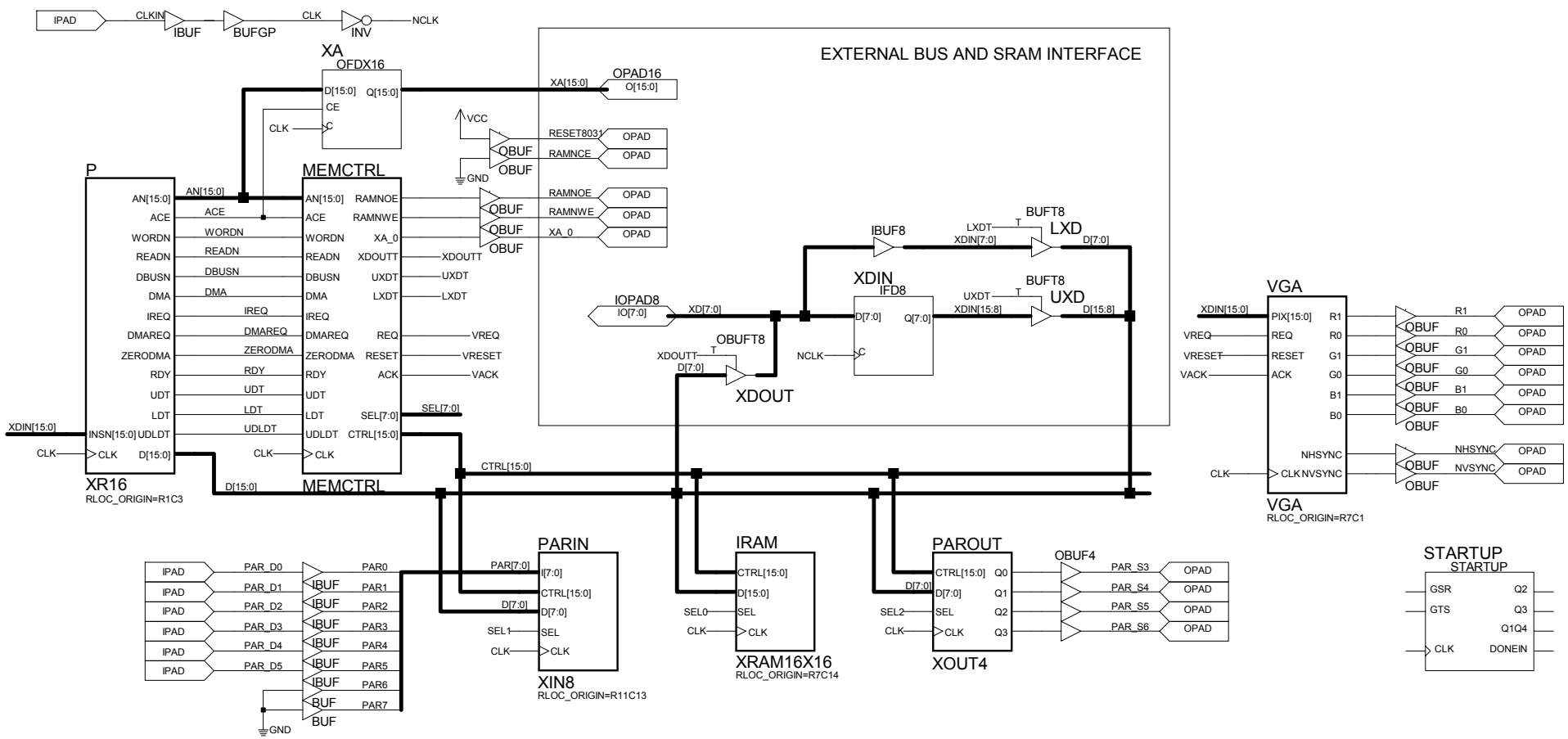


Figure 2: XSOC Floorplan

4 Revision History

Version	Date	Description
0.10	2000/2/20	First draft
0.11	2000/3/20	Noted XA[16:15] connection on 128 KB XS40+ boards



Project: XSOC	
Sheet: XSOC	
Date: 06/07/99	

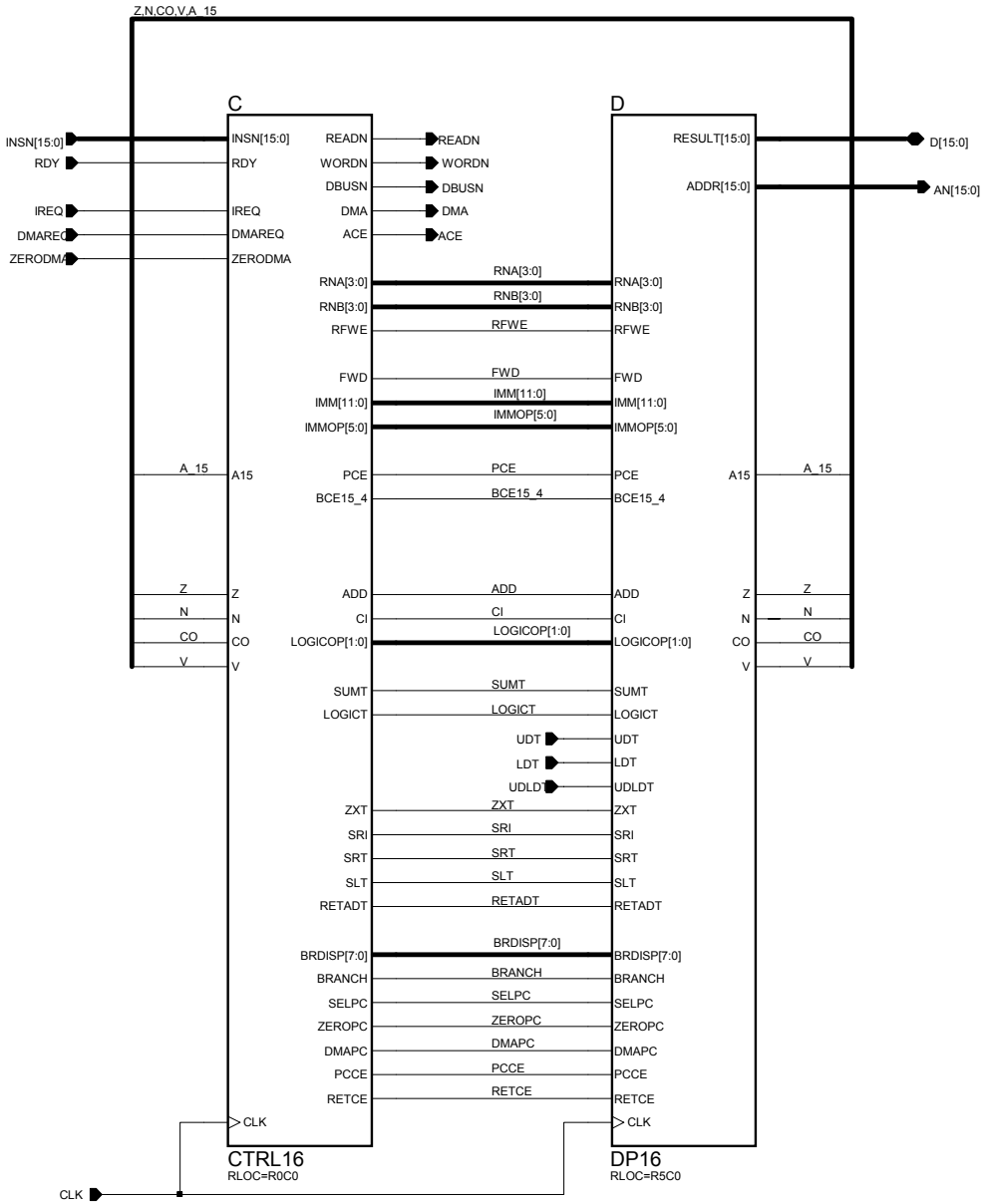


Figure S2: XR16 CPU Schematic

	Project: [None]
	Macro: XR16
	Date: 02/23/100

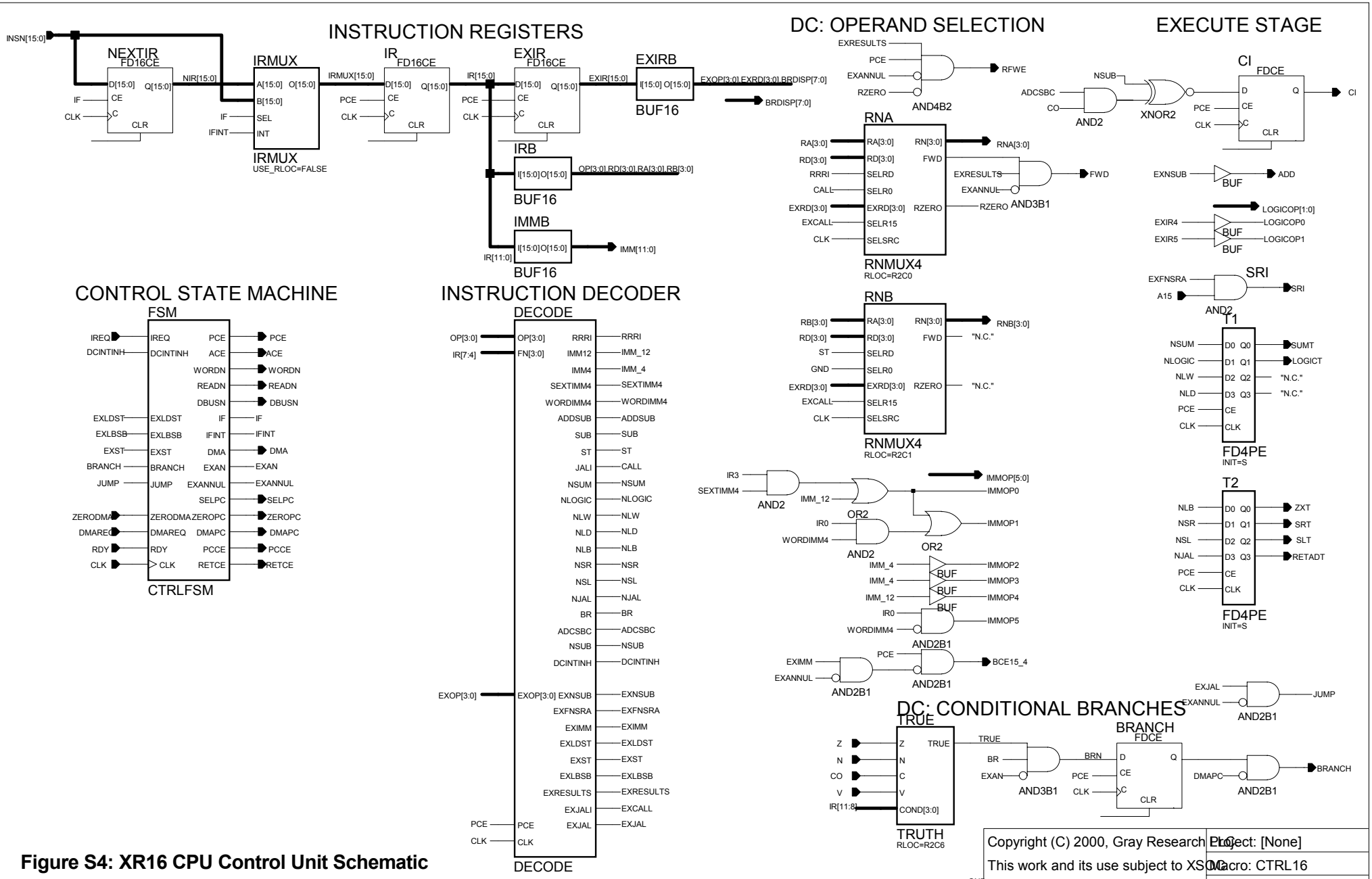
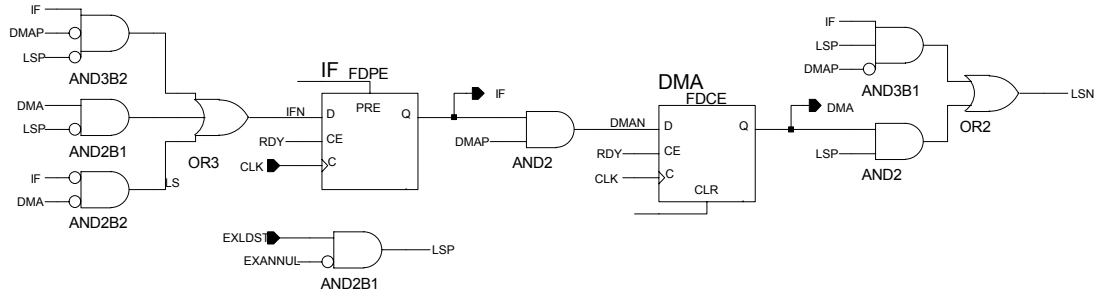


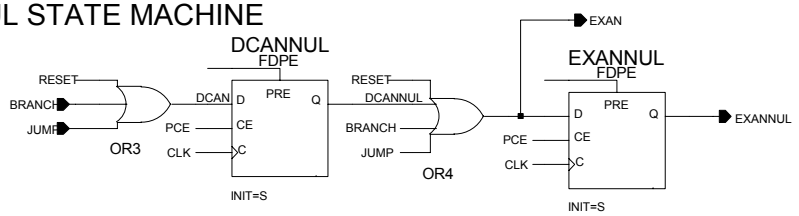
Figure S4: XR16 CPU Control Unit Schematic

Project: [None]	
Macro: CTRL16	
Date: 02/23/100	

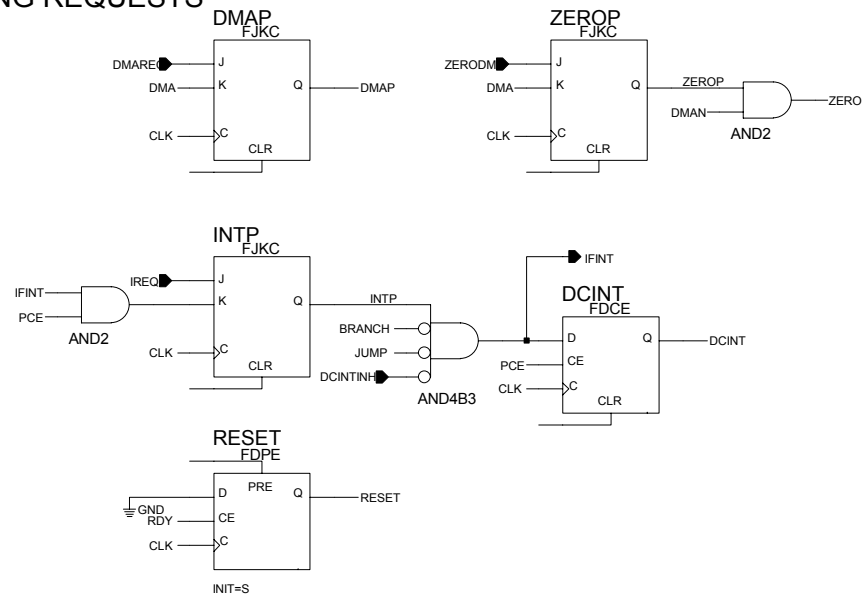
MEM CYCLE STATE MACHINE



ANNUL STATE MACHINE



PENDING REQUESTS



FSM OUTPUTS

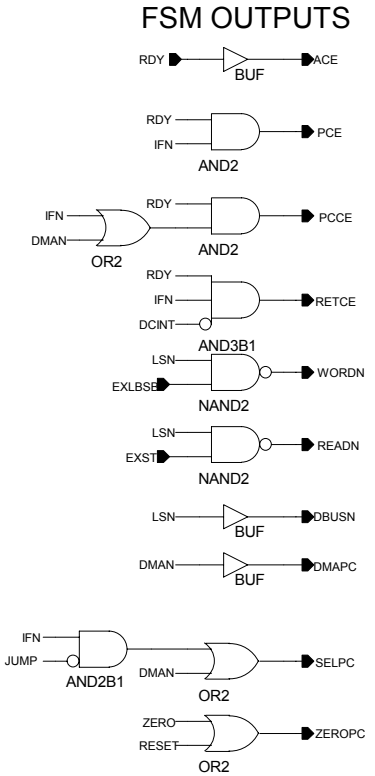


Figure S5: XR16 CPU Control Unit State Machines

Project: [None]	
Macro: CTRLFSM	
Date: 02/23/100	

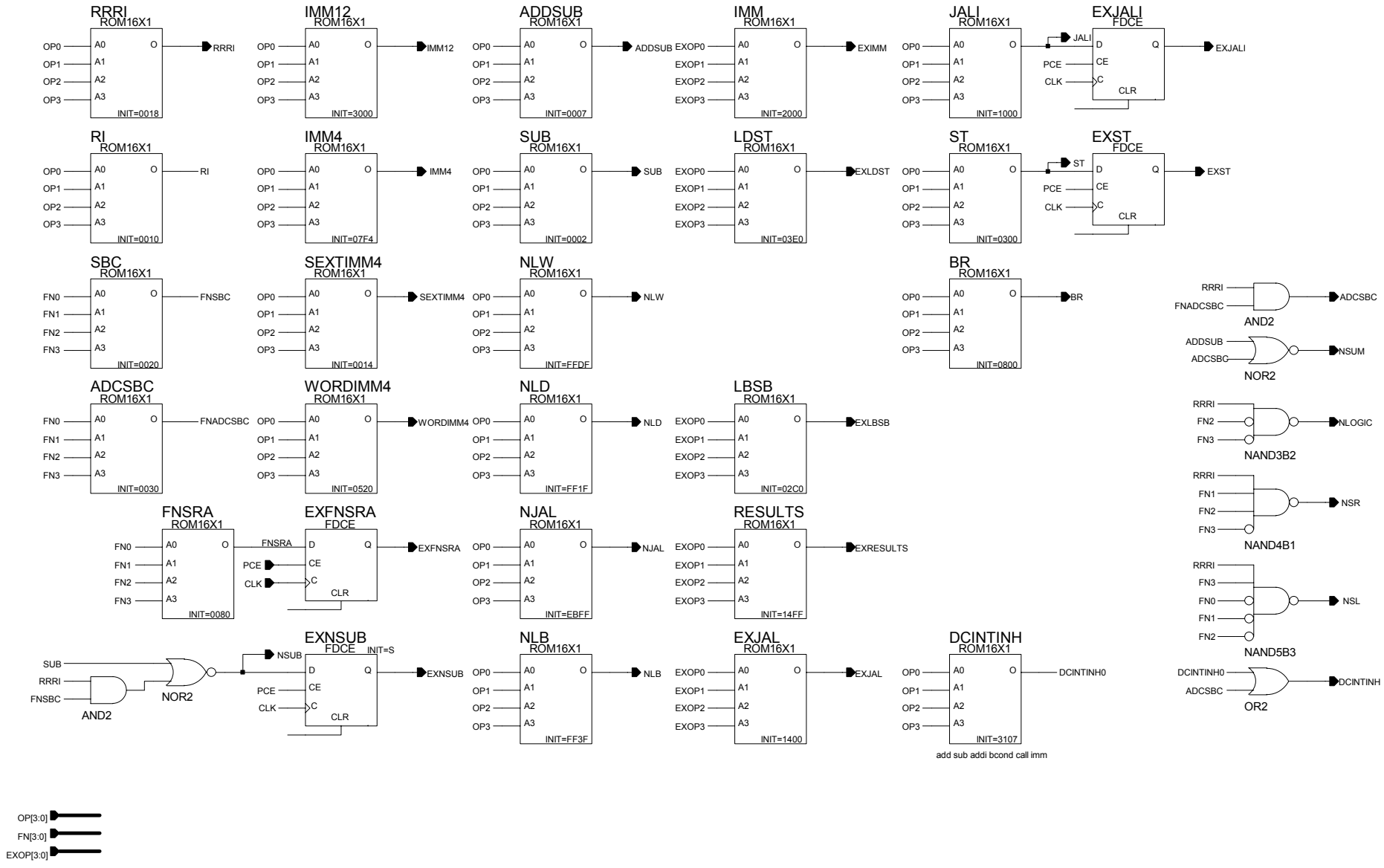
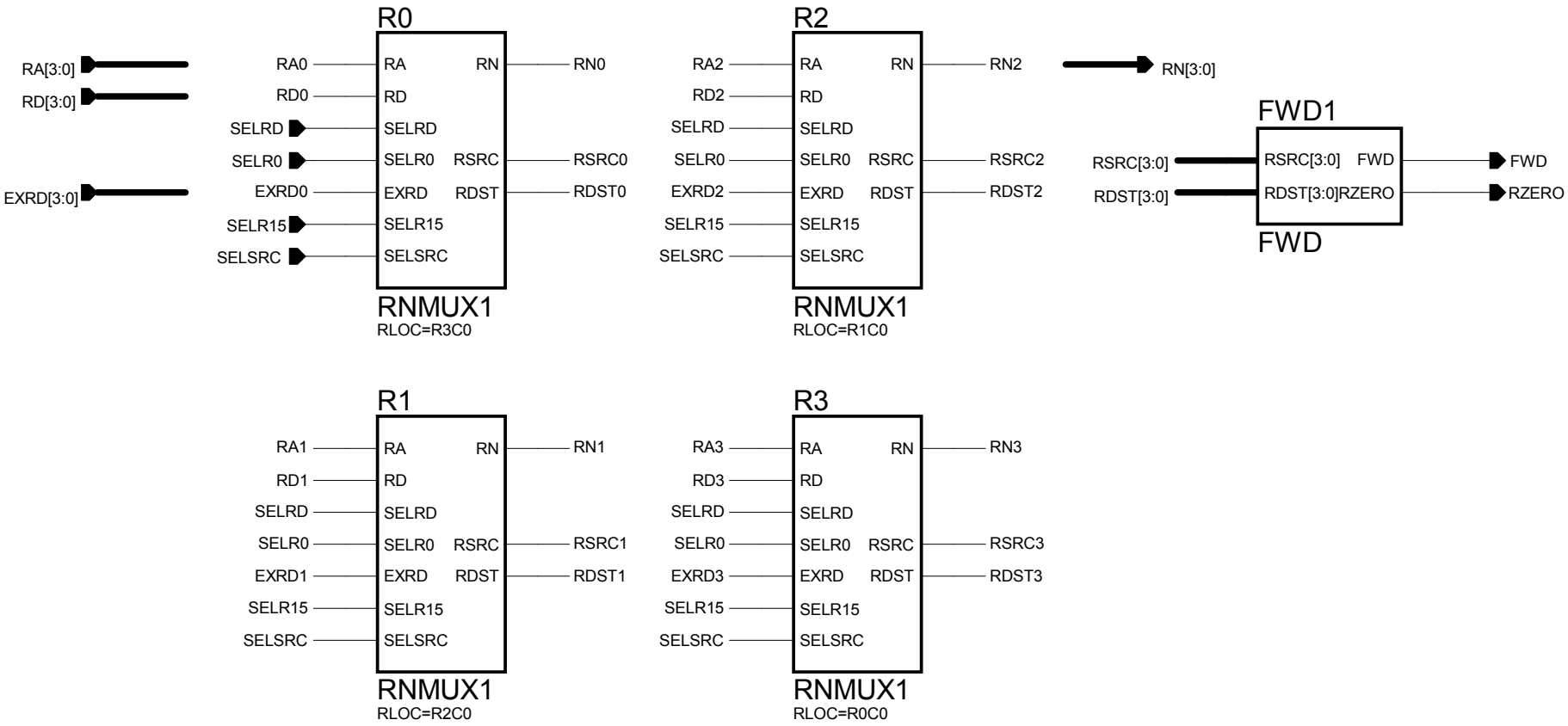


Figure S6: XR16 Instruction Decoder

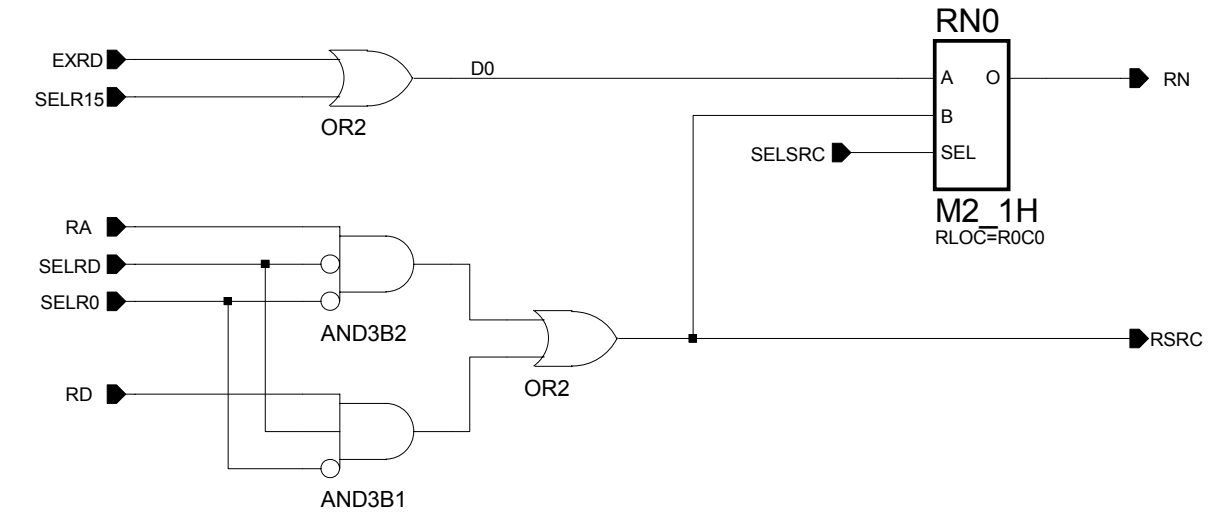
Project: [None]	
Macro: DECODE	
Date: 02/23/100	

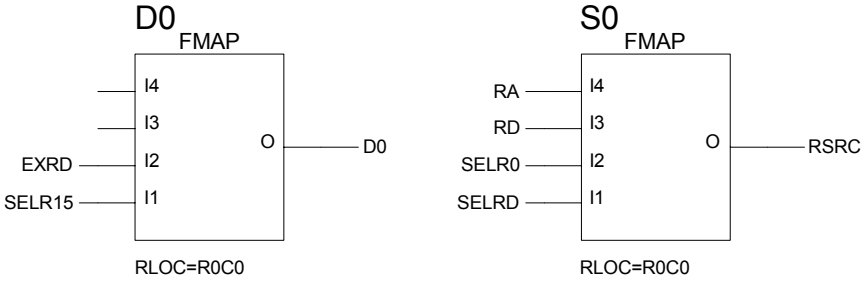


Project: [None]

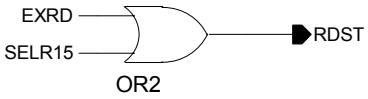
Macro: RNMUX4

Date: 02/23/100





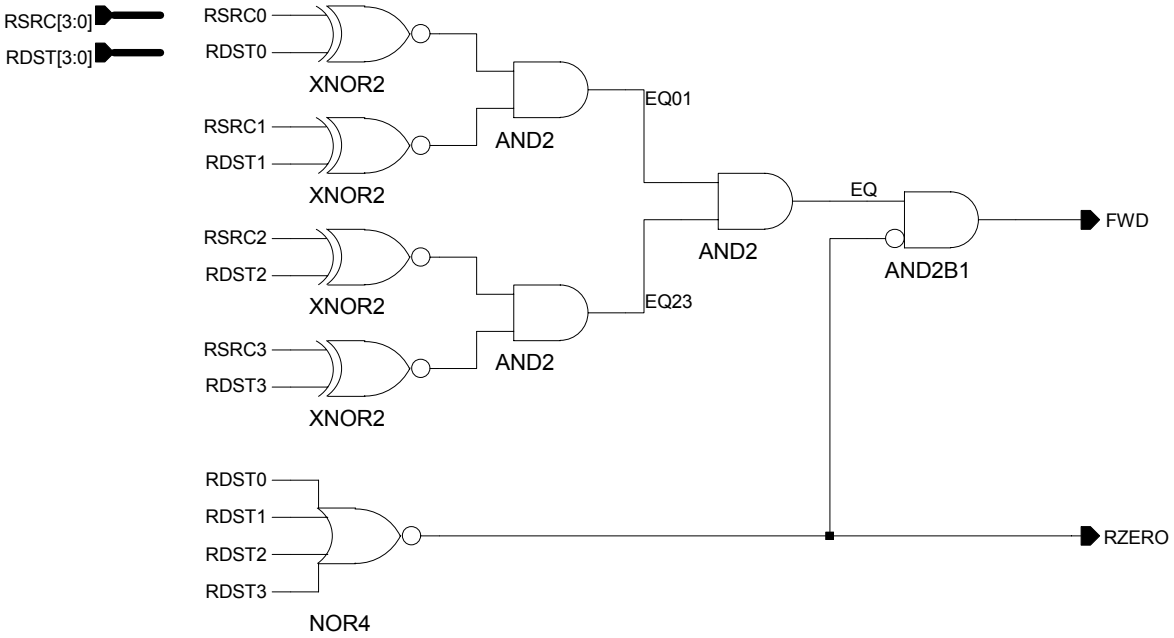
"Copy of RDST circuit; CLBs only get 2 combinatorial outputs"



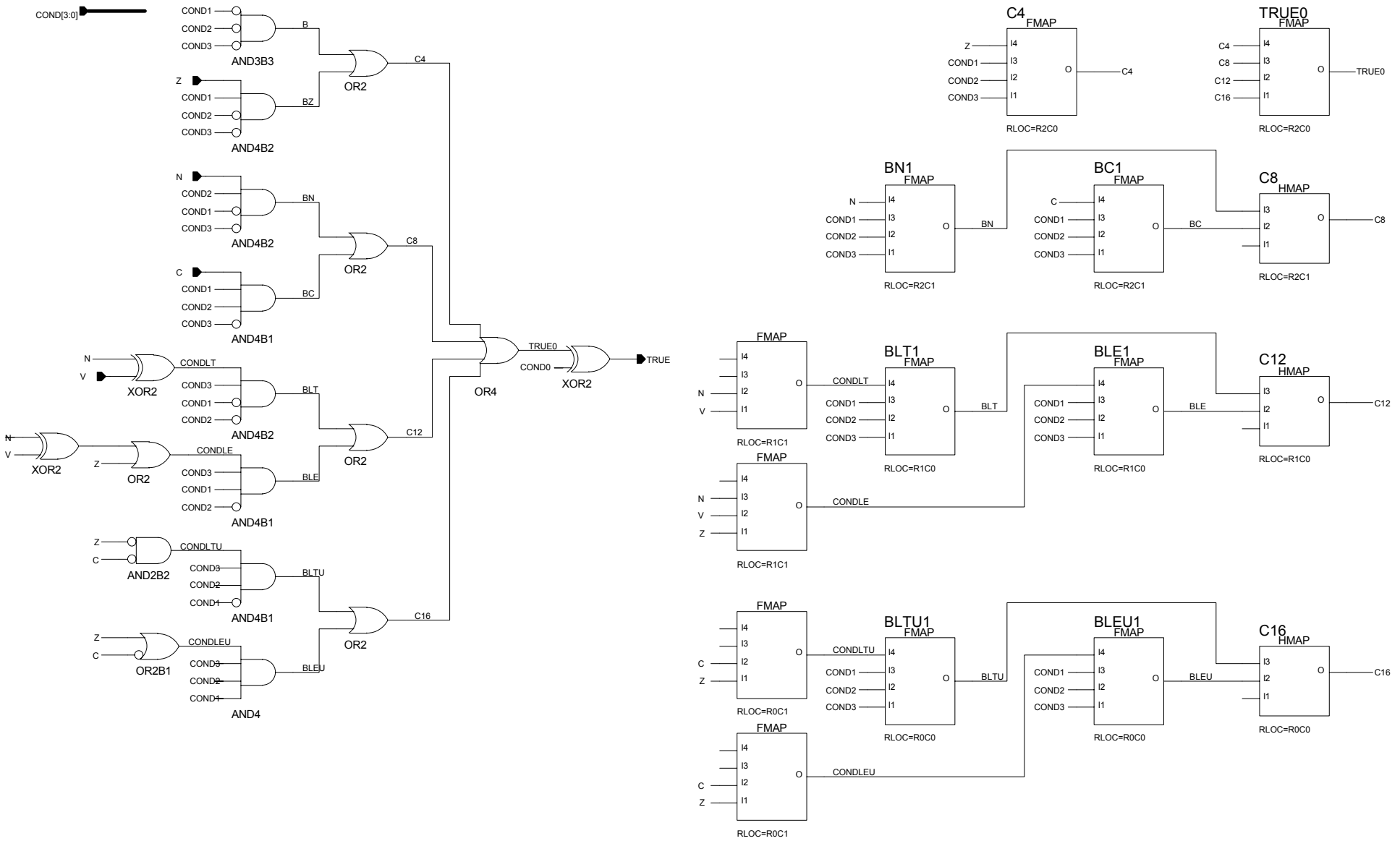
Project: [None]

Macro: RNMUX1

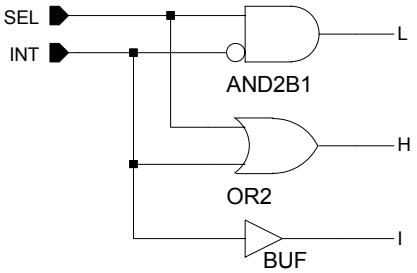
Date: 02/23/100

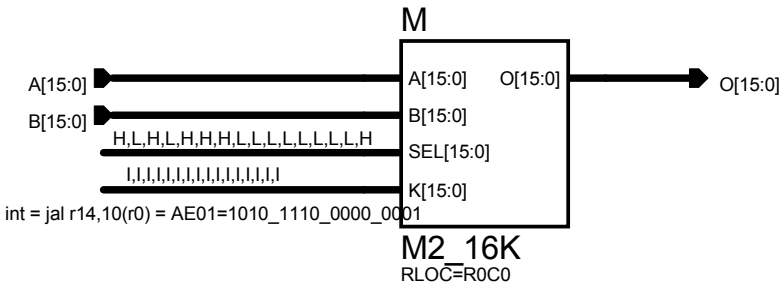


	Project: [None]
	Macro: FWD
	Date: 02/23/100



Project: [None]	
Macro: TRUTH	
Date: 02/23/100	





Project: [None]

Macro: IRMUX

Date: 02/23/100

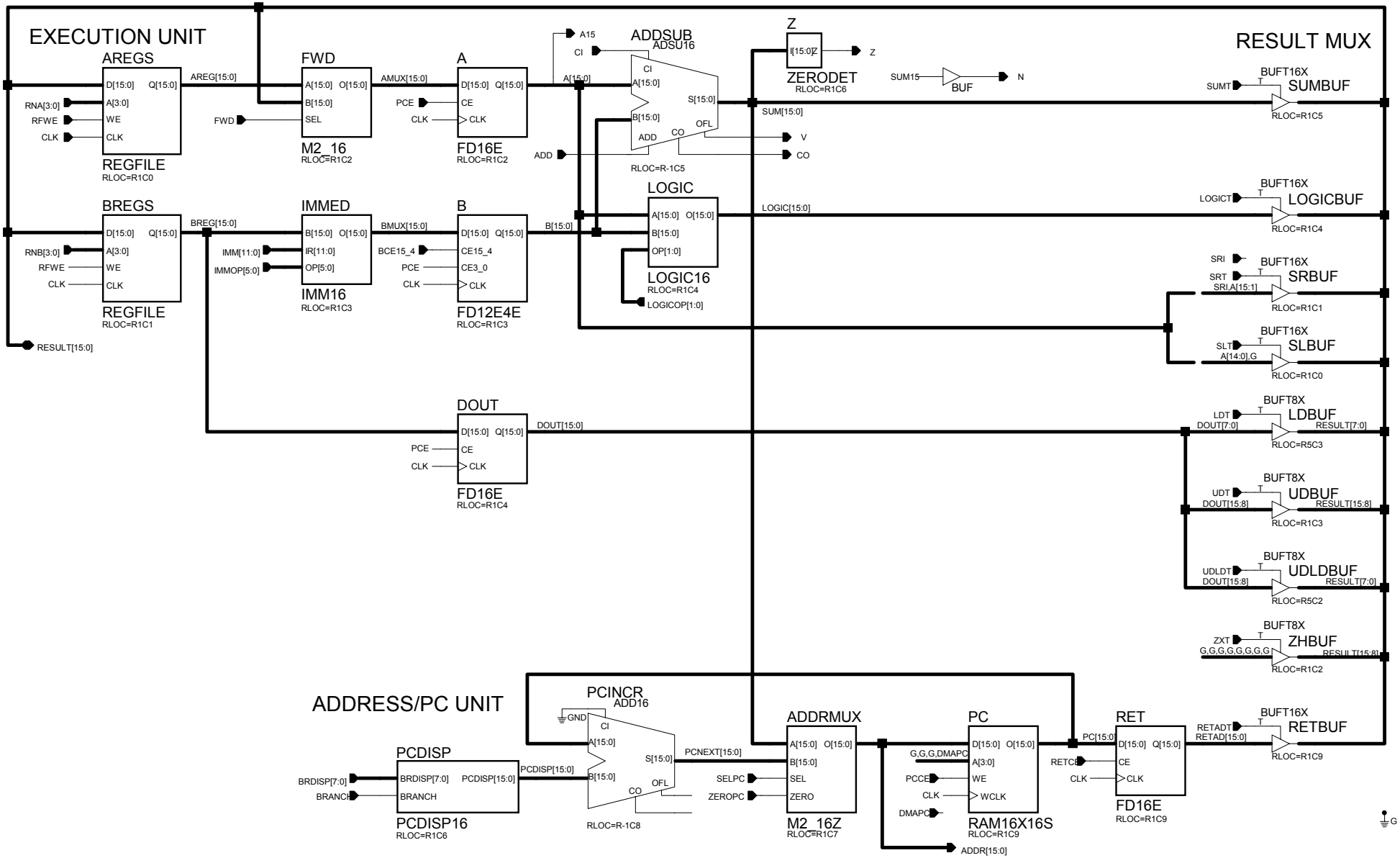
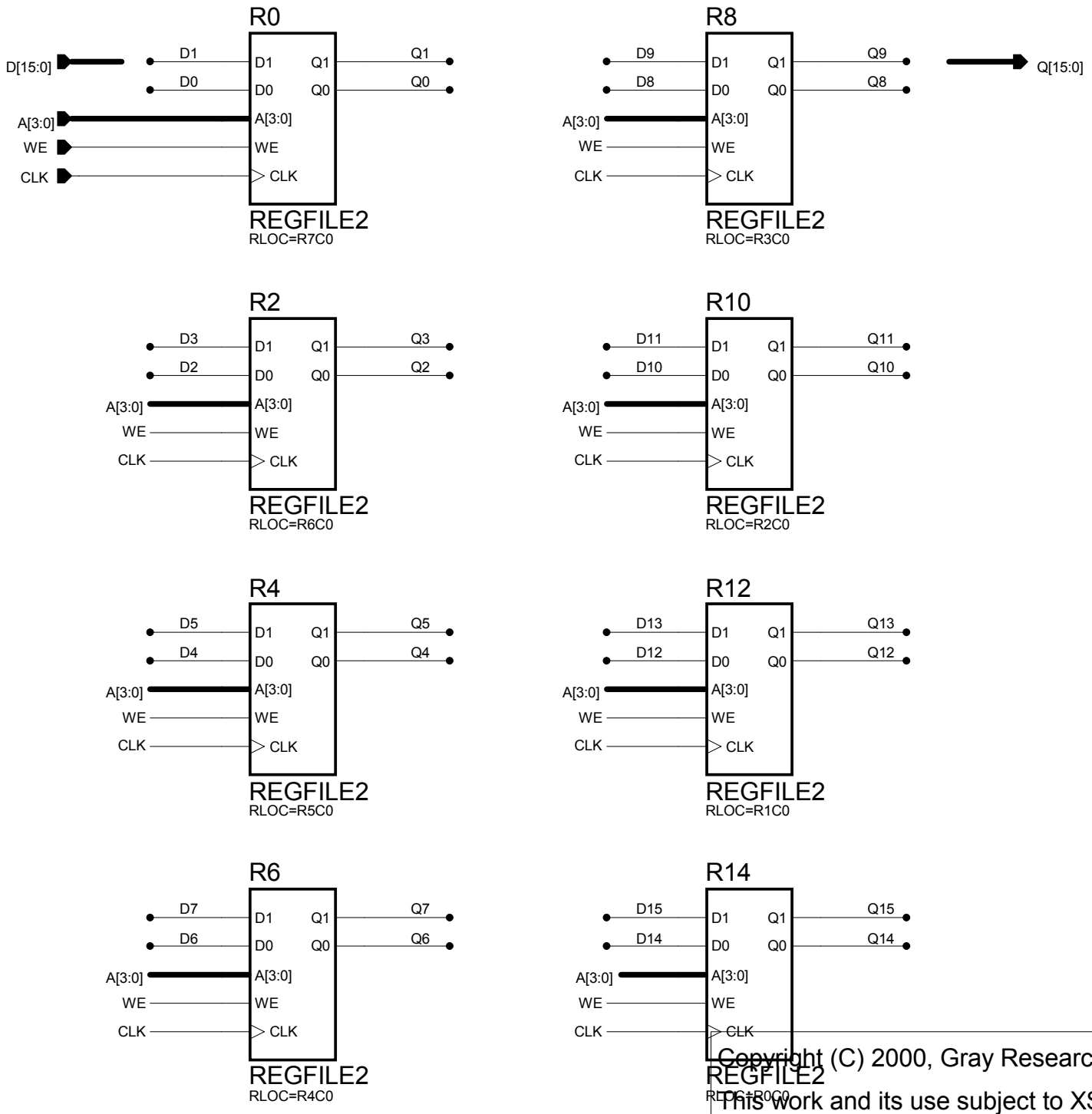
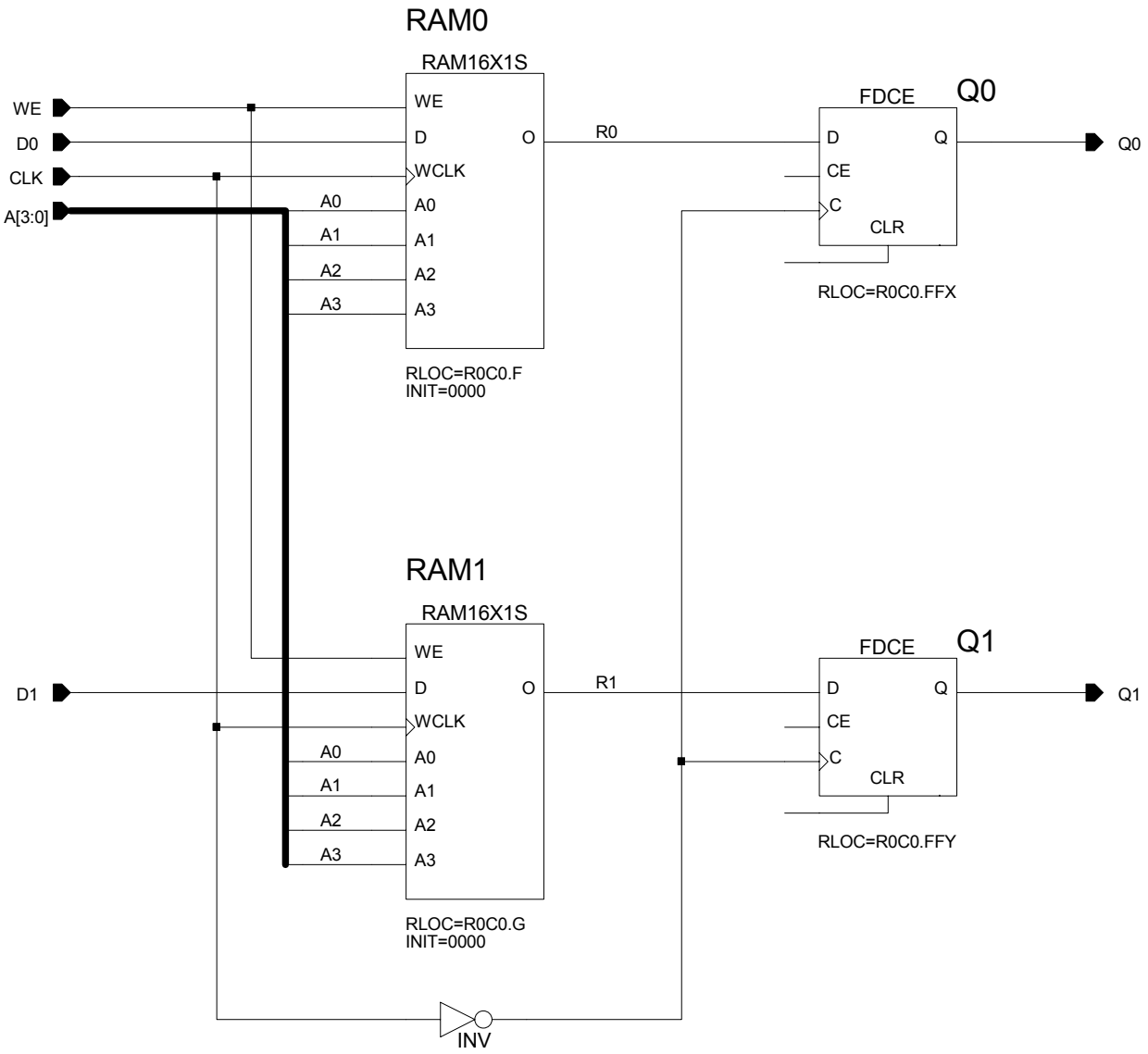


Figure S3: XR16 CPU Datapath Schematic

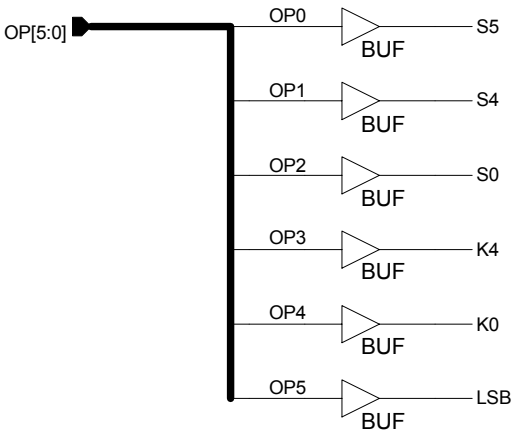
Project: [None]
Macro: DP16
Date: 02/23/100

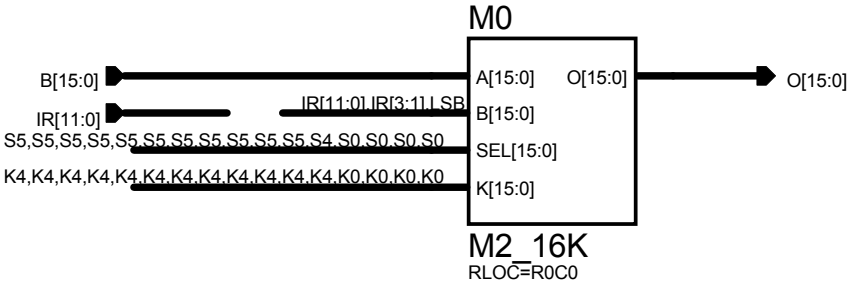


Project: [None]
Macro: REGFILE
Date: 02/23/100

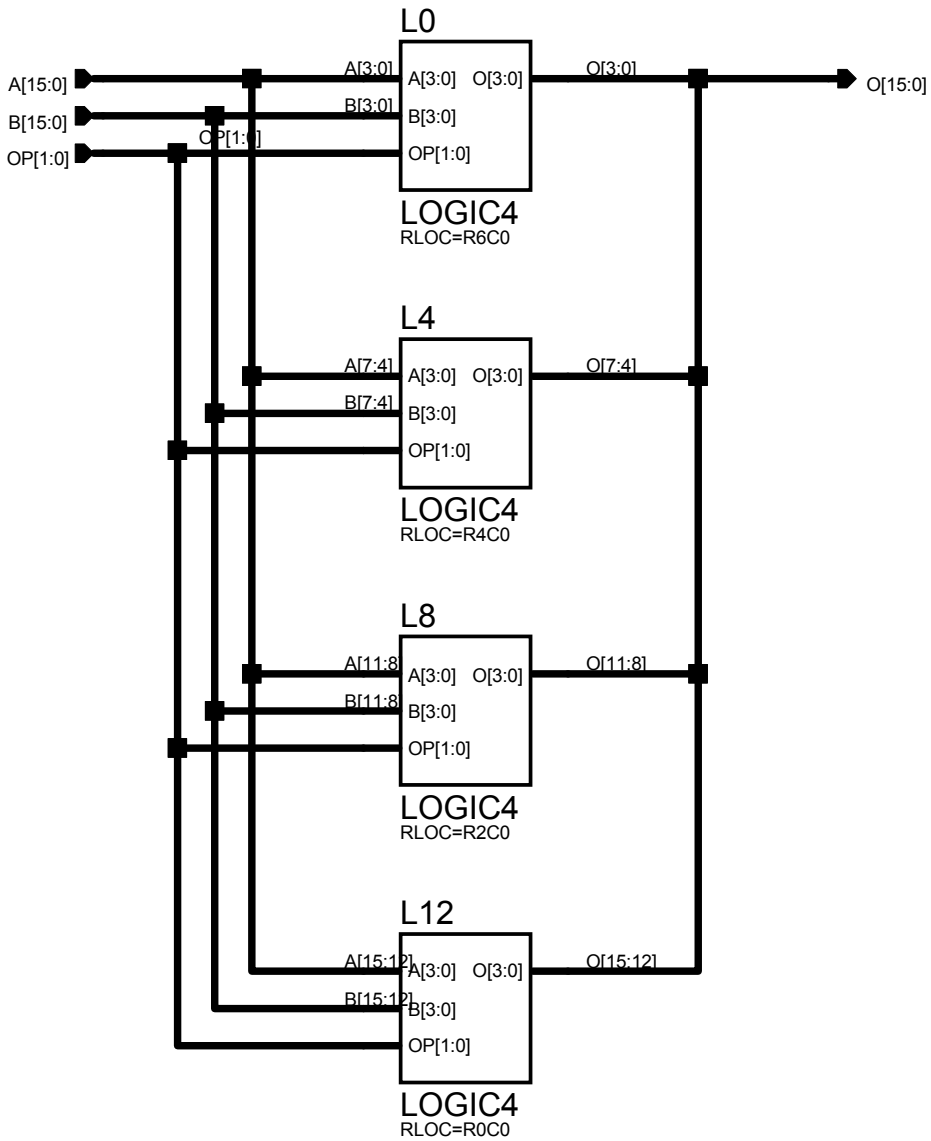


Project: [None]	
Macro: REGFILE2	
Date: 02/23/100	

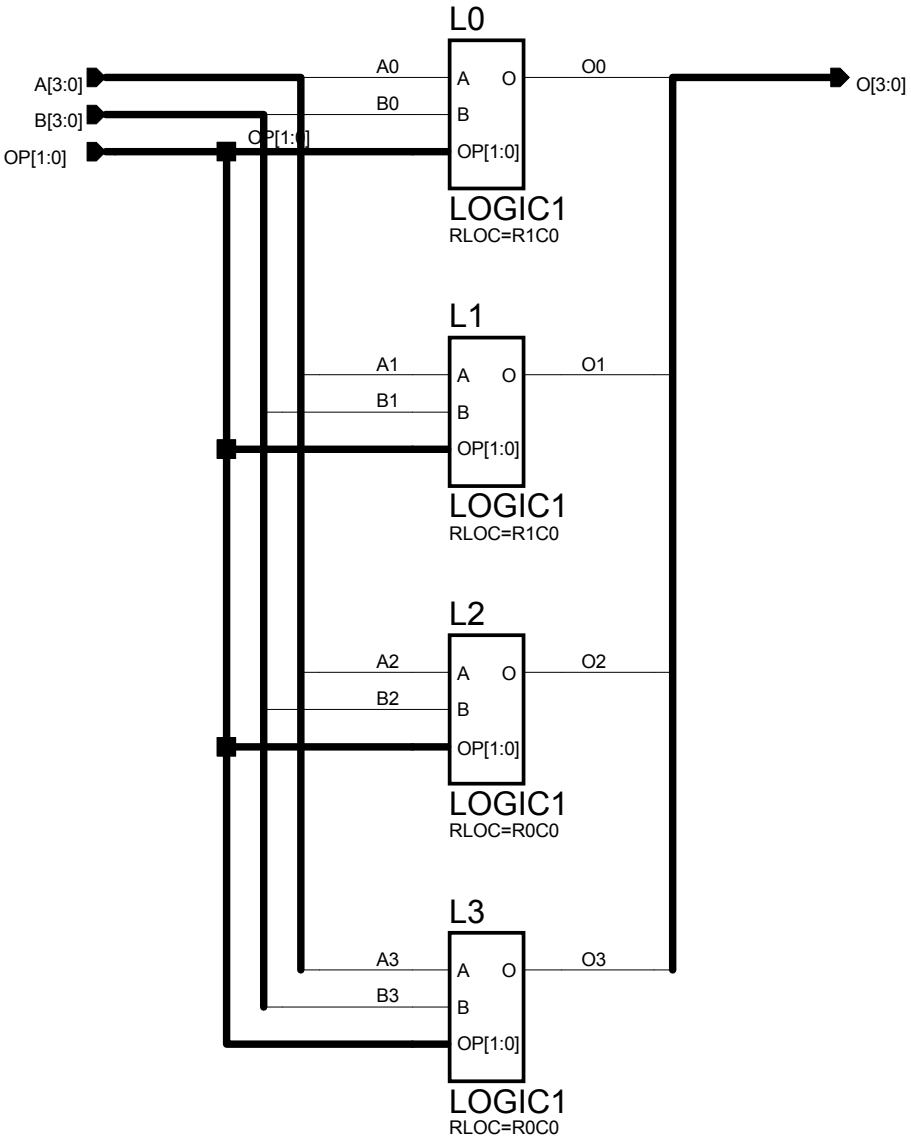




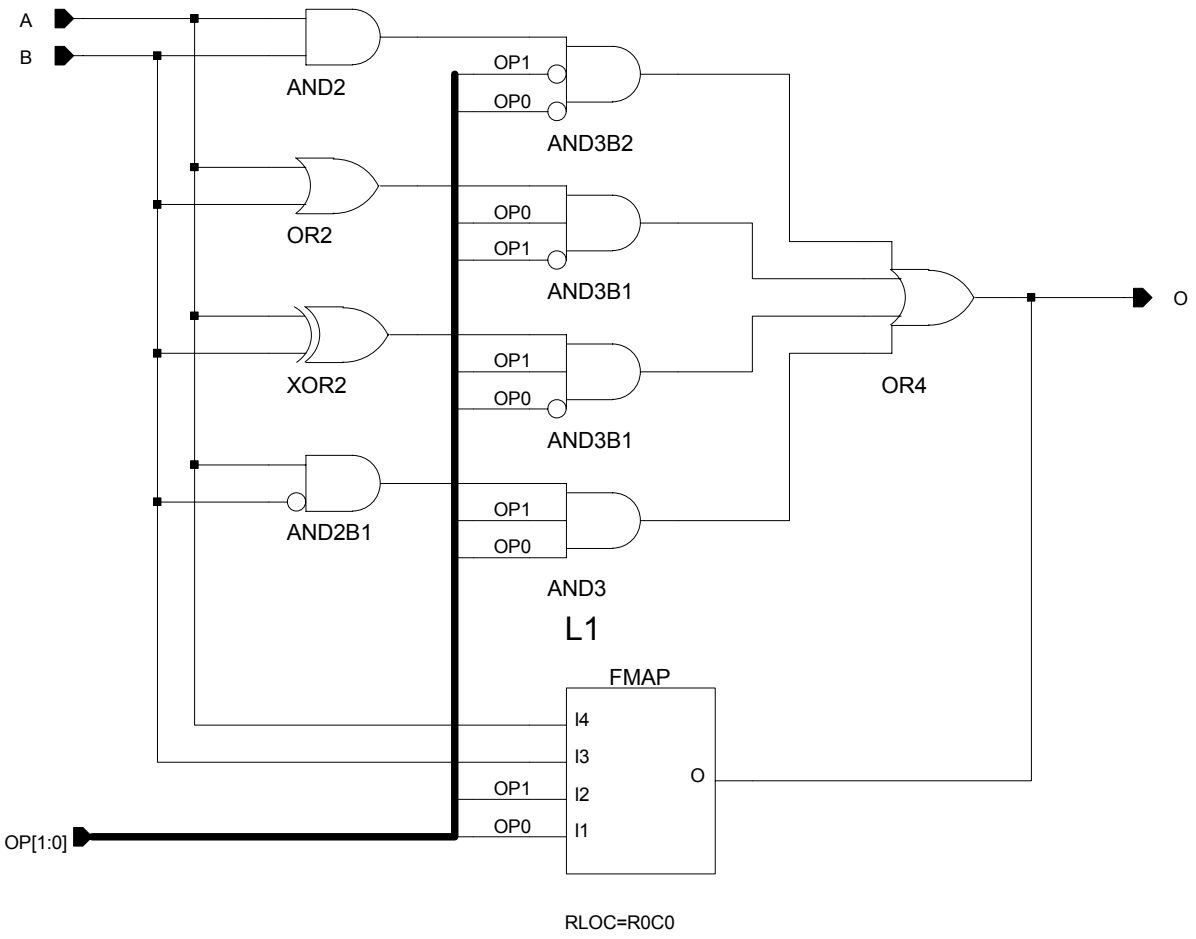
	Project: [None]
	Macro: IMM16
	Date: 02/23/100



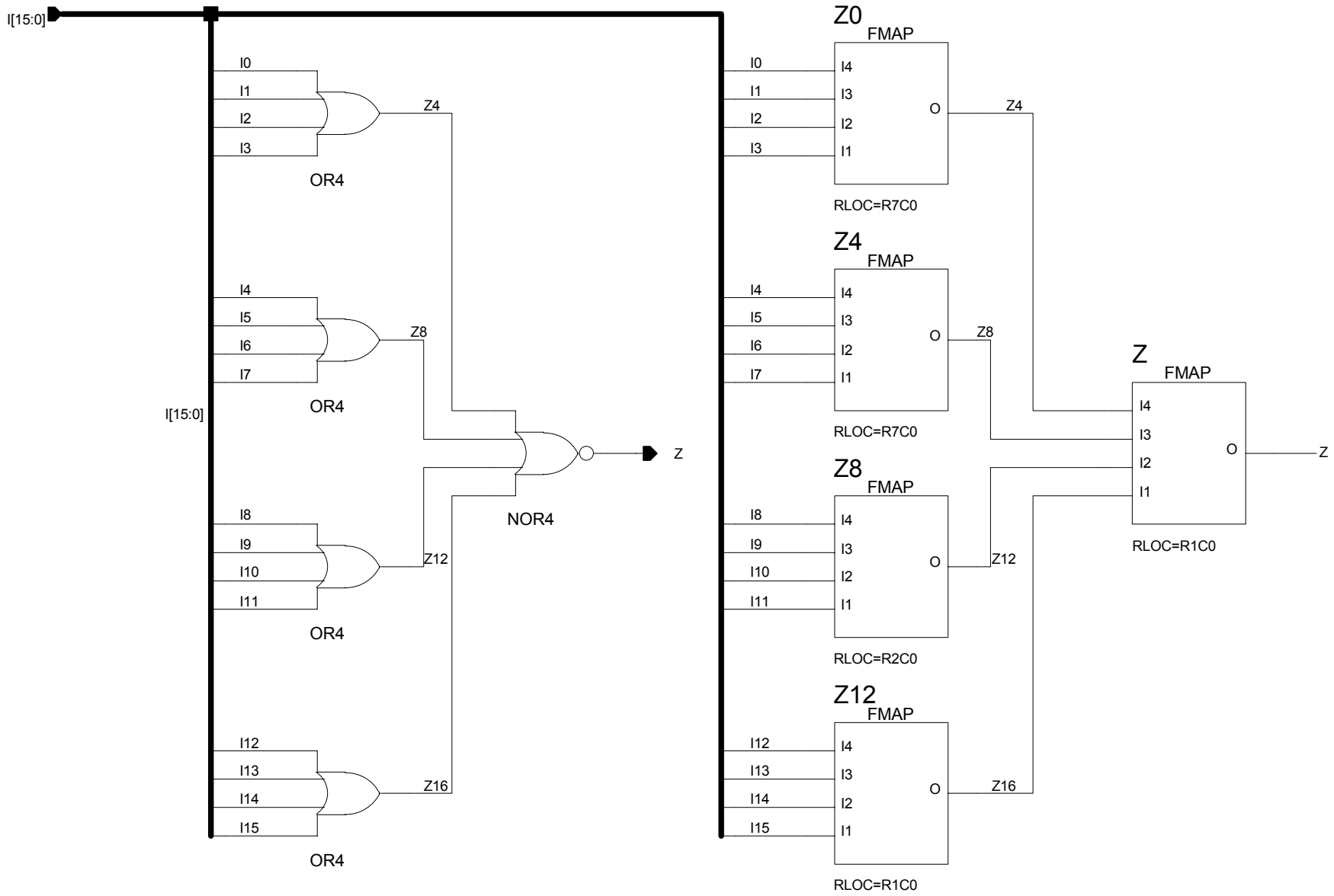
	Project: [None]
	Macro: LOGIC16
	Date: 02/23/100



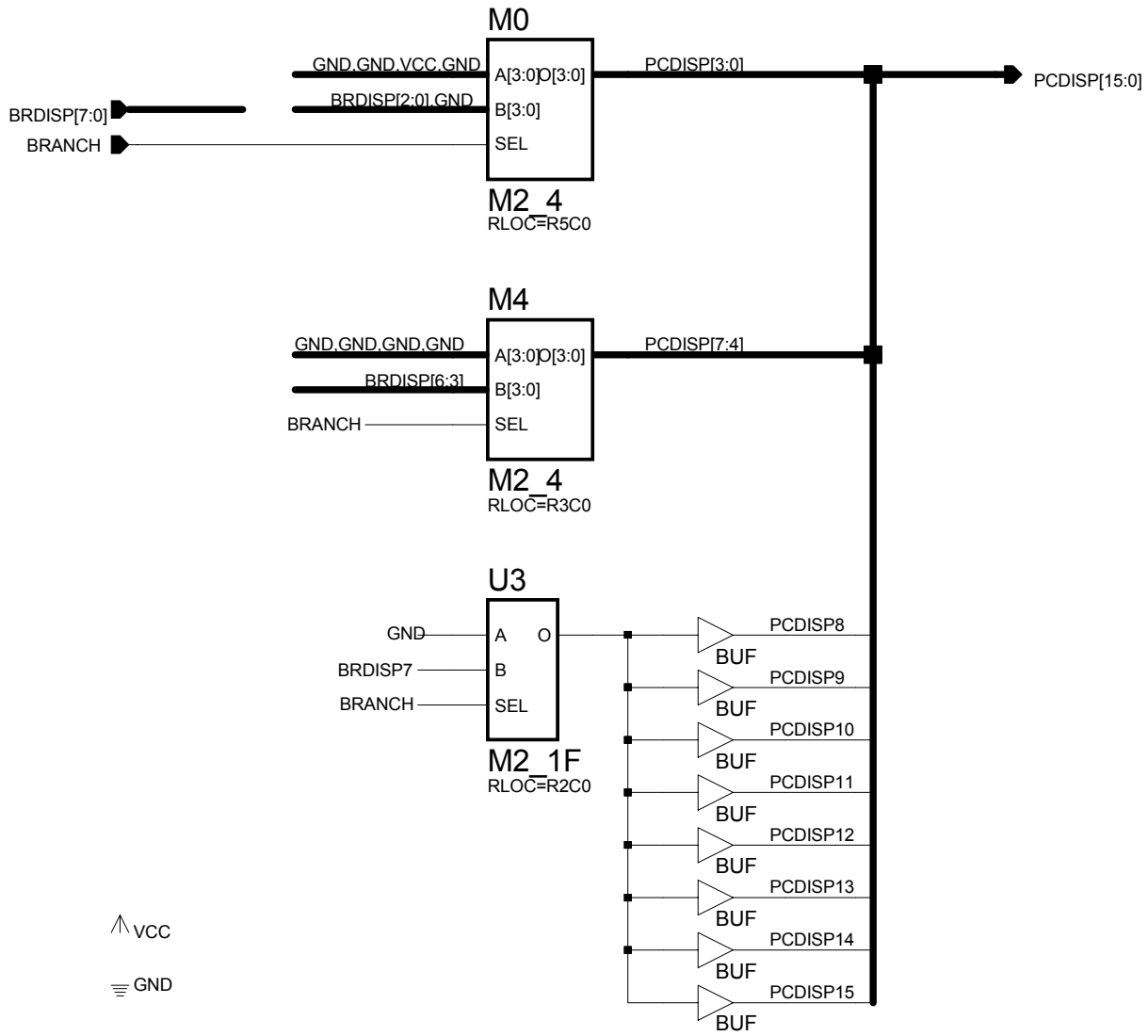
	Project: [None]
	Macro: LOGIC4
	Date: 02/23/100



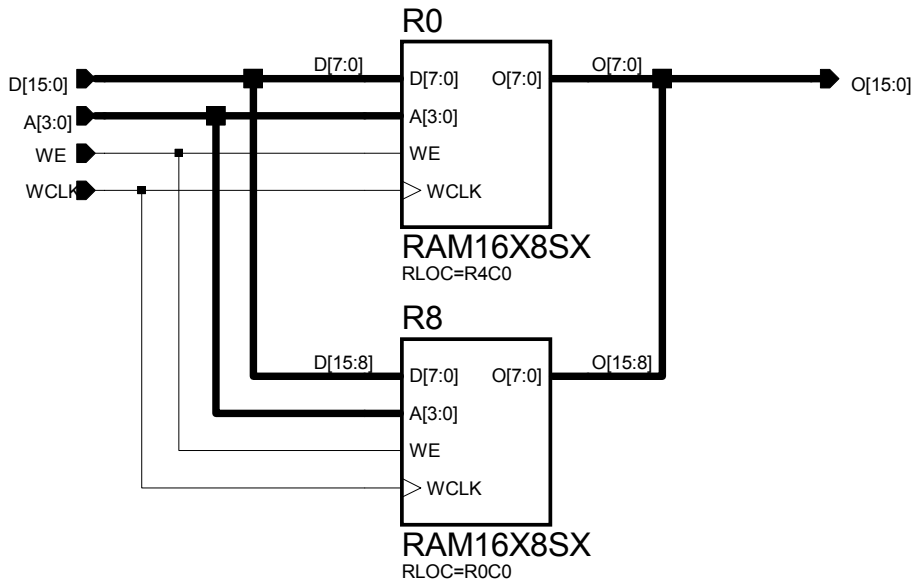
	Project: [None]
	Macro: LOGIC1
	Date: 02/23/100



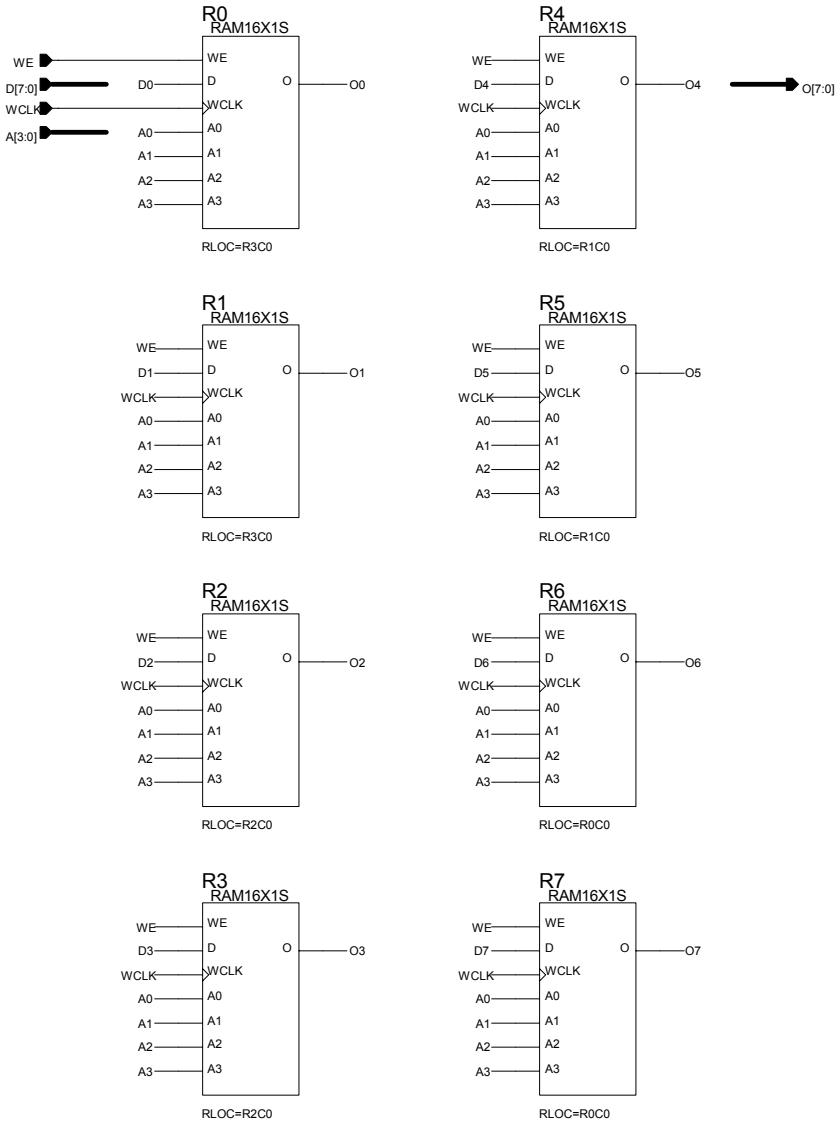
Project: [None]
Macro: ZERODET
Date: 02/23/100



	Project: [None]
	Macro: PCDISP16
	Date: 02/23/100



	Project: [None]
	Macro: RAM16X16S
	Date: 02/23/100



Project: [None]	
Macro: RAM16X8SX	
Date: 02/23/100	

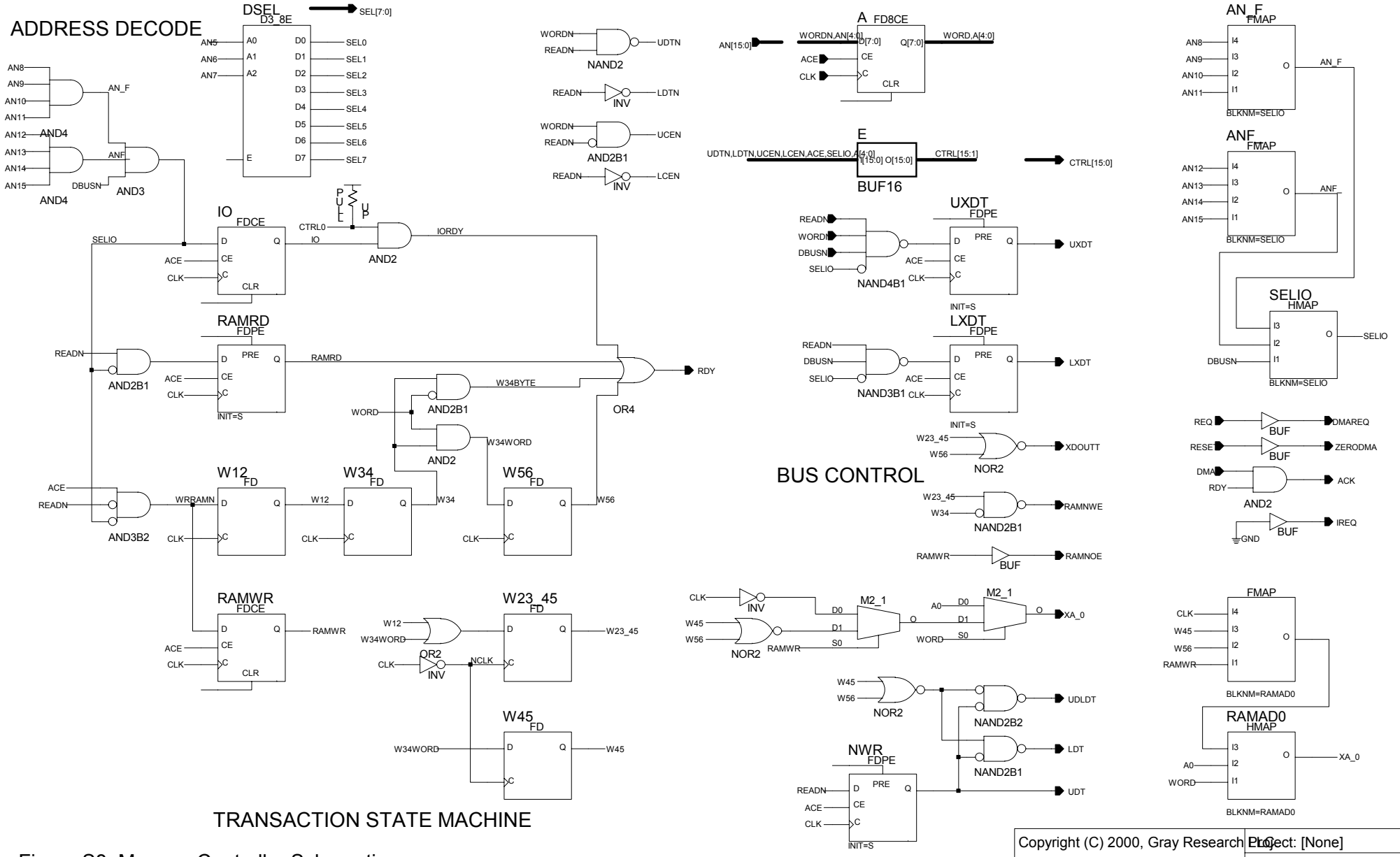


Figure S8: Memory Controller Schematic

Project: [None]	
Macro: MEMCTRL	
Date: 02/23/100	

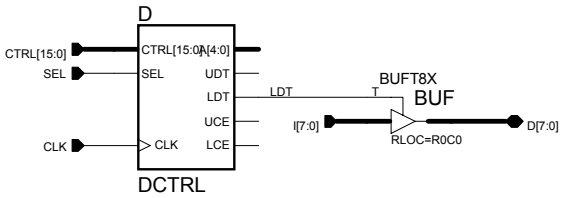
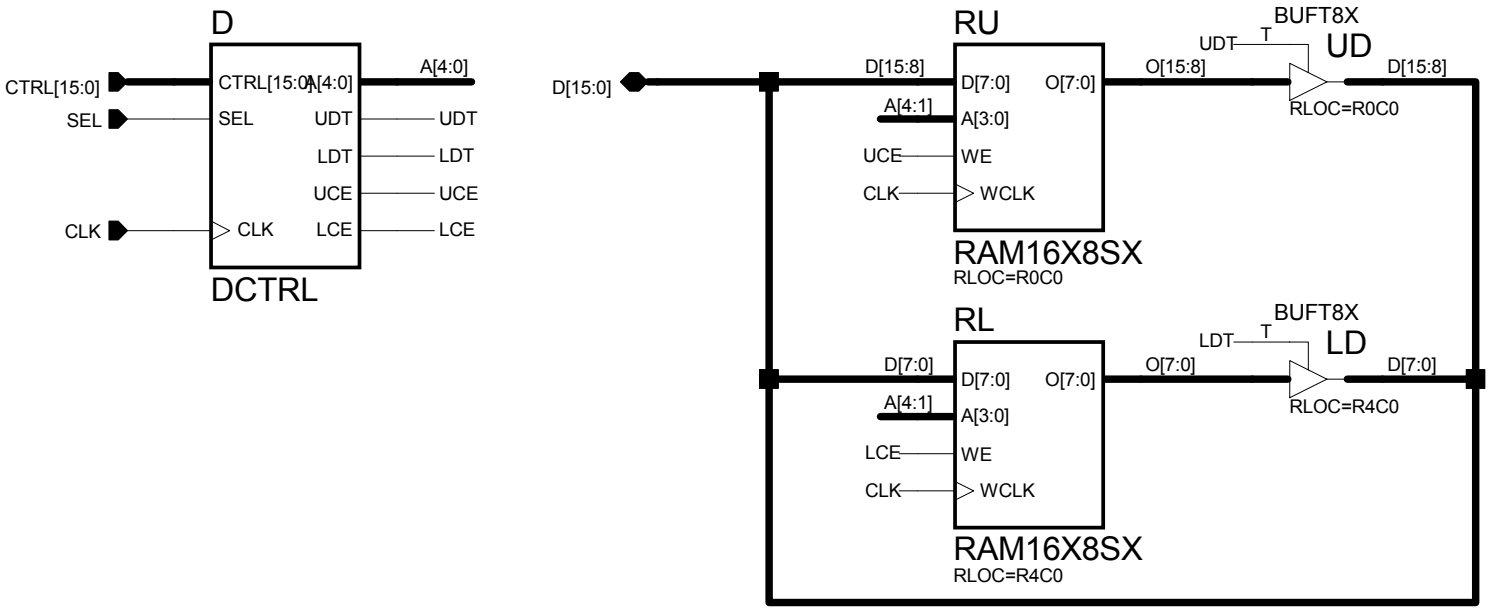
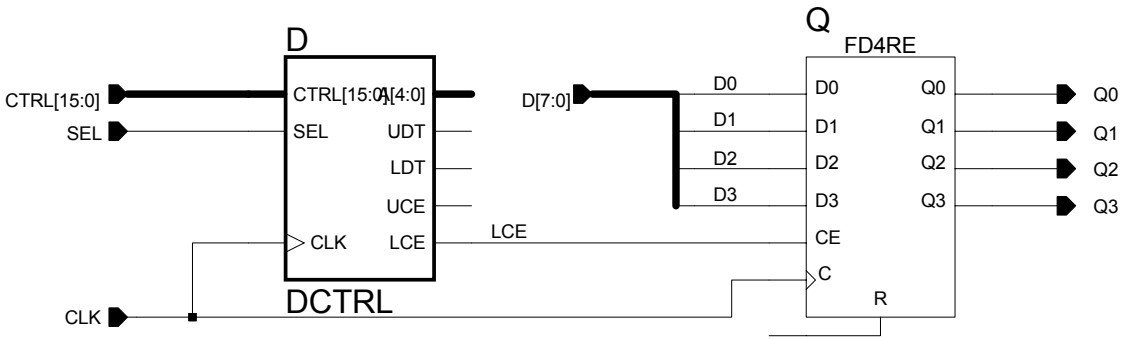


Figure S7: 8-bit Input Peripheral Schematic

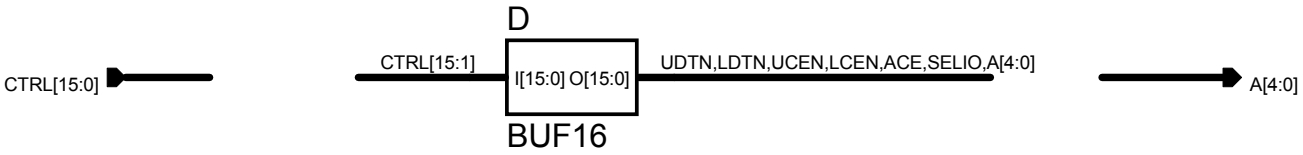
Project: [None]	Page: 13
Macro: XIN8	
Date: 02/23/100	

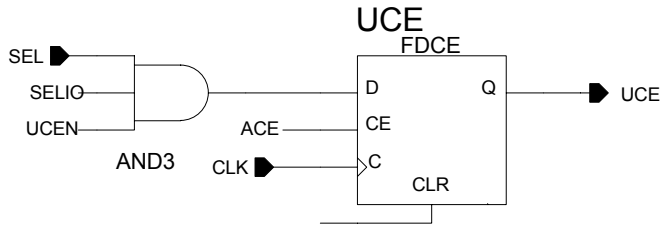


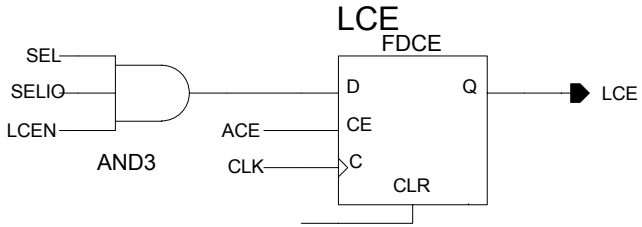
	Project: [None]
	Macro: XRAM16X16
	Date: 02/23/100

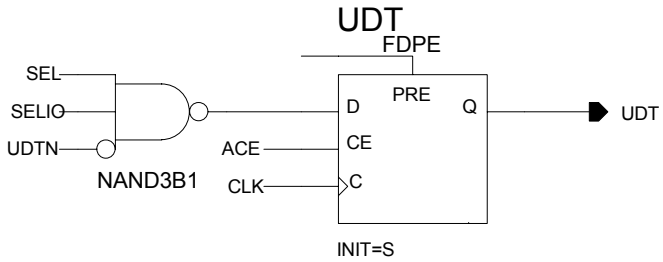


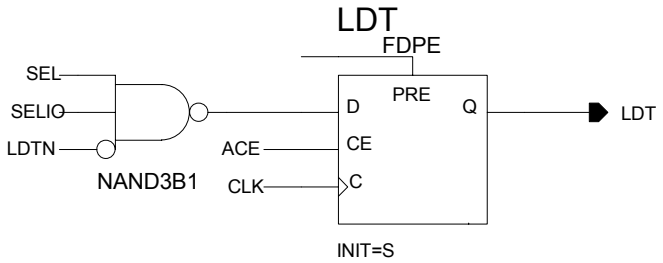
	Project: [None]
	Macro: XOUT4
	Date: 02/23/100











Project: [None]
Macro: DCTRL
Date: 02/23/100

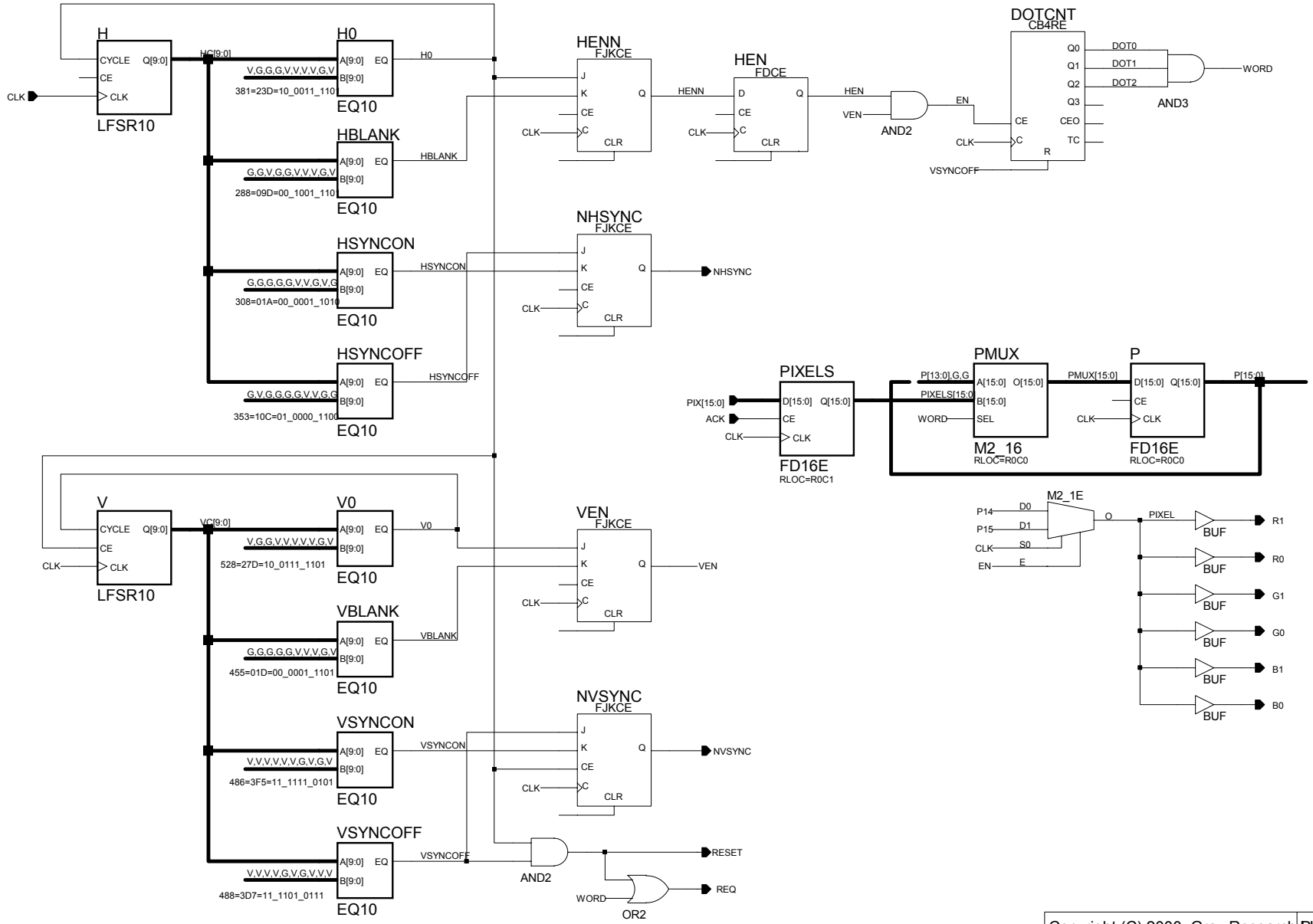
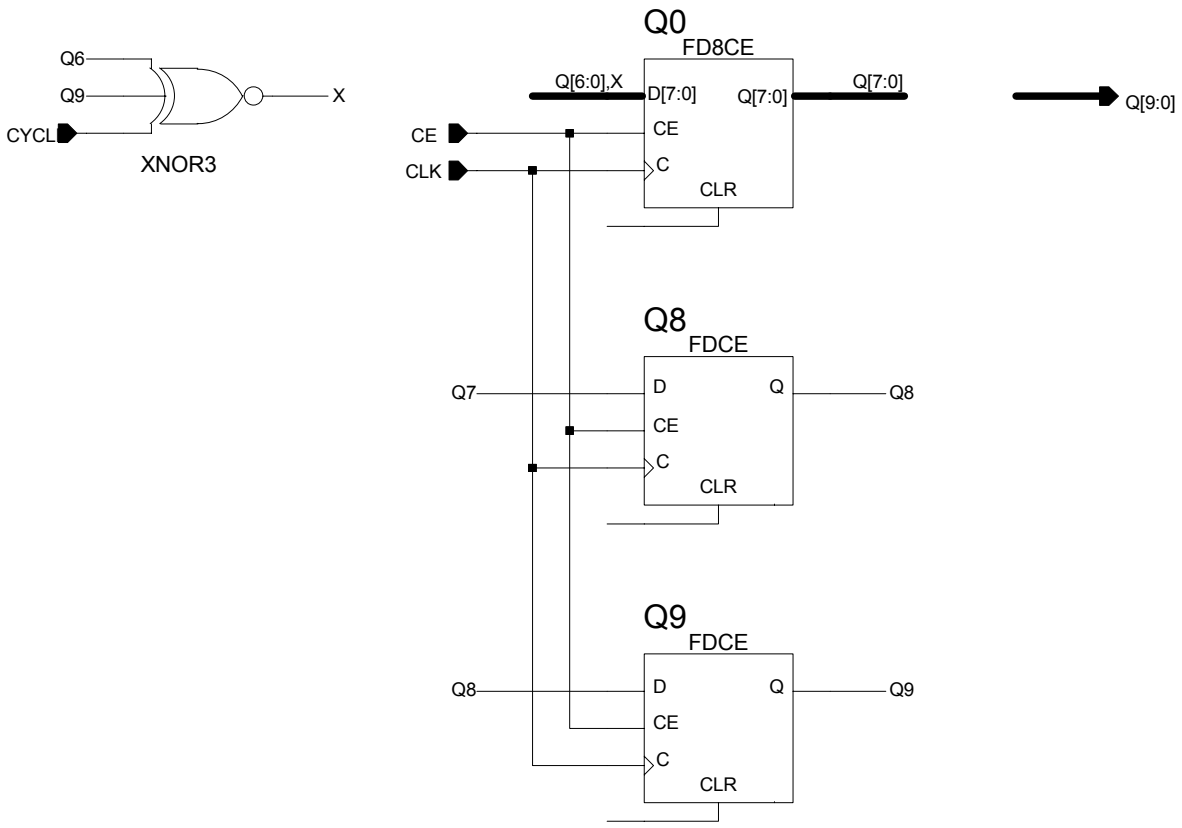
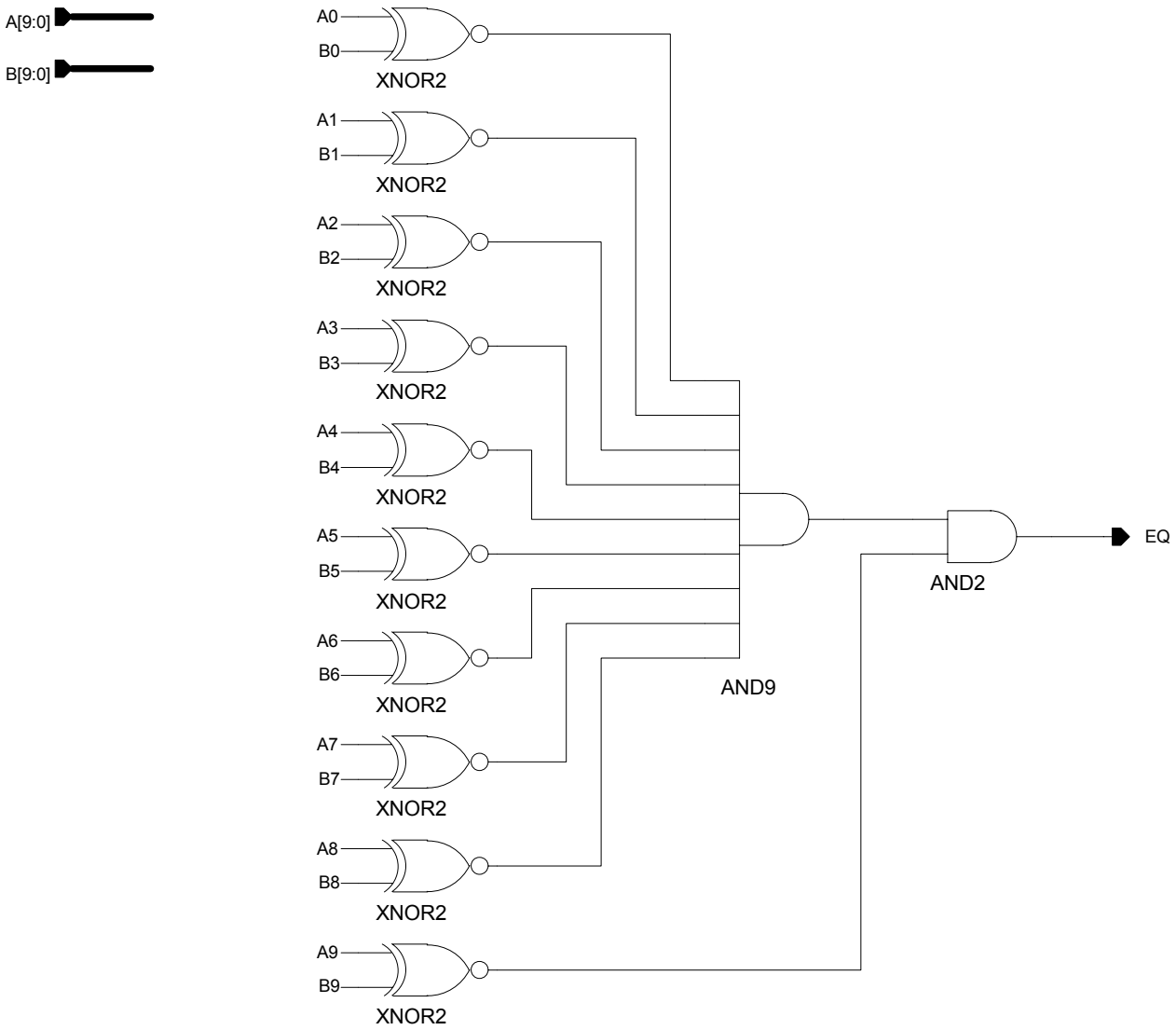


Figure S9: Video Controller Schematic

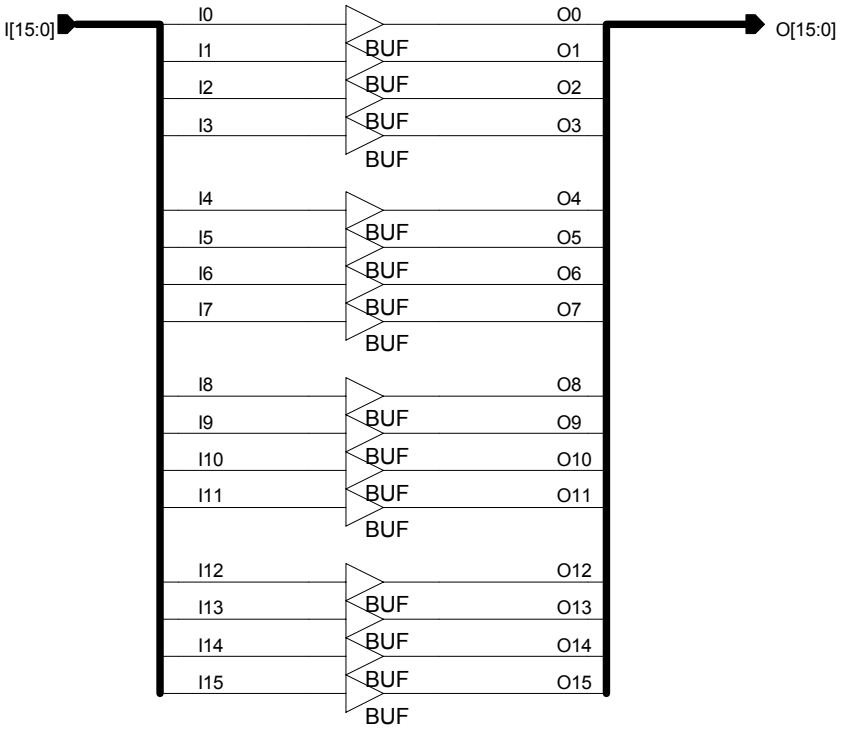
Project: [None]	
Macro: VGA	
Date: 02/23/100	



	Project: [None]
	Macro: LFSR10
	Date: 02/23/100



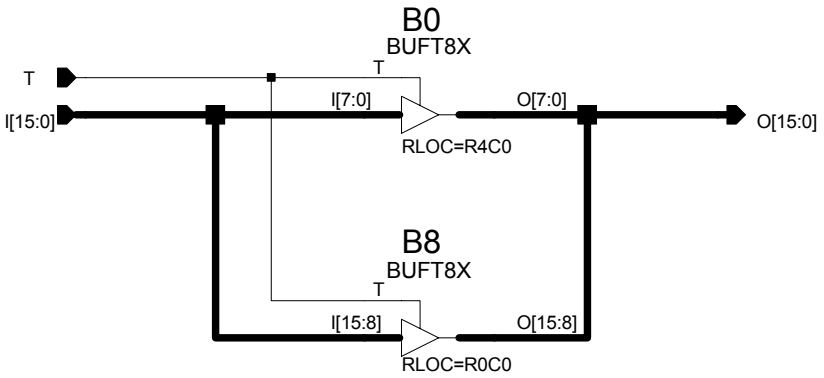
	Project: [None]
	Macro: EQ10
	Date: 02/23/100



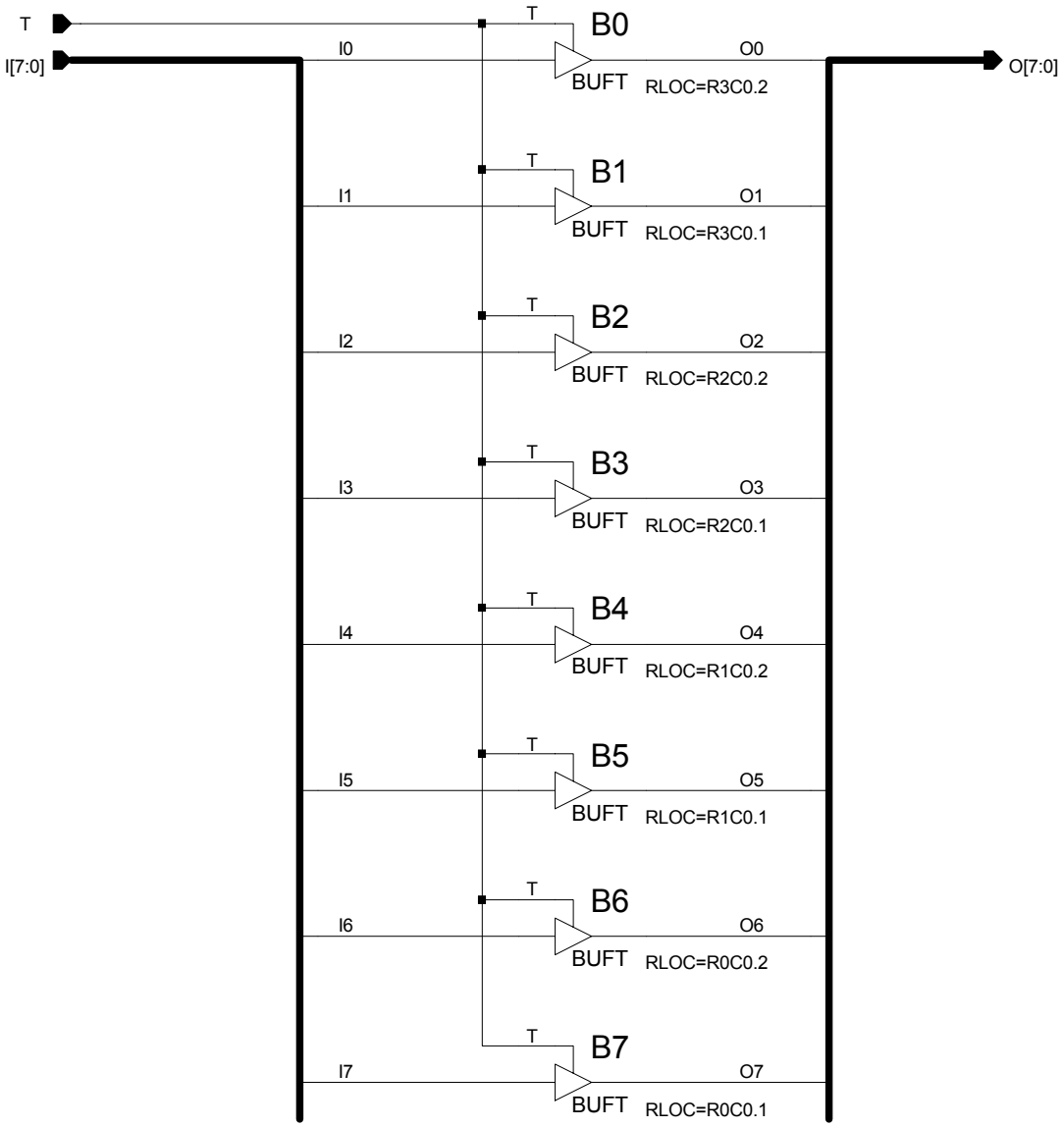
Project: [None]

Macro: BUF16

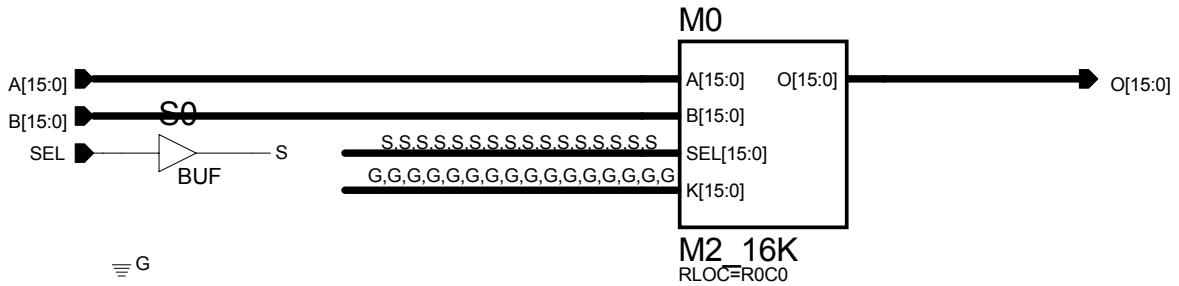
Date: 02/23/100

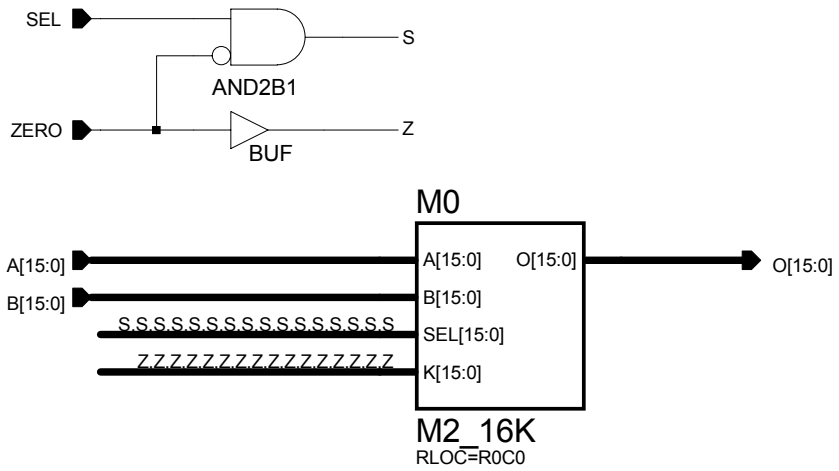


Project: [None]
Macro: BUFT16X
Date: 02/23/100



Project: [None]
Macro: BUFT8X
Date: 02/23/100

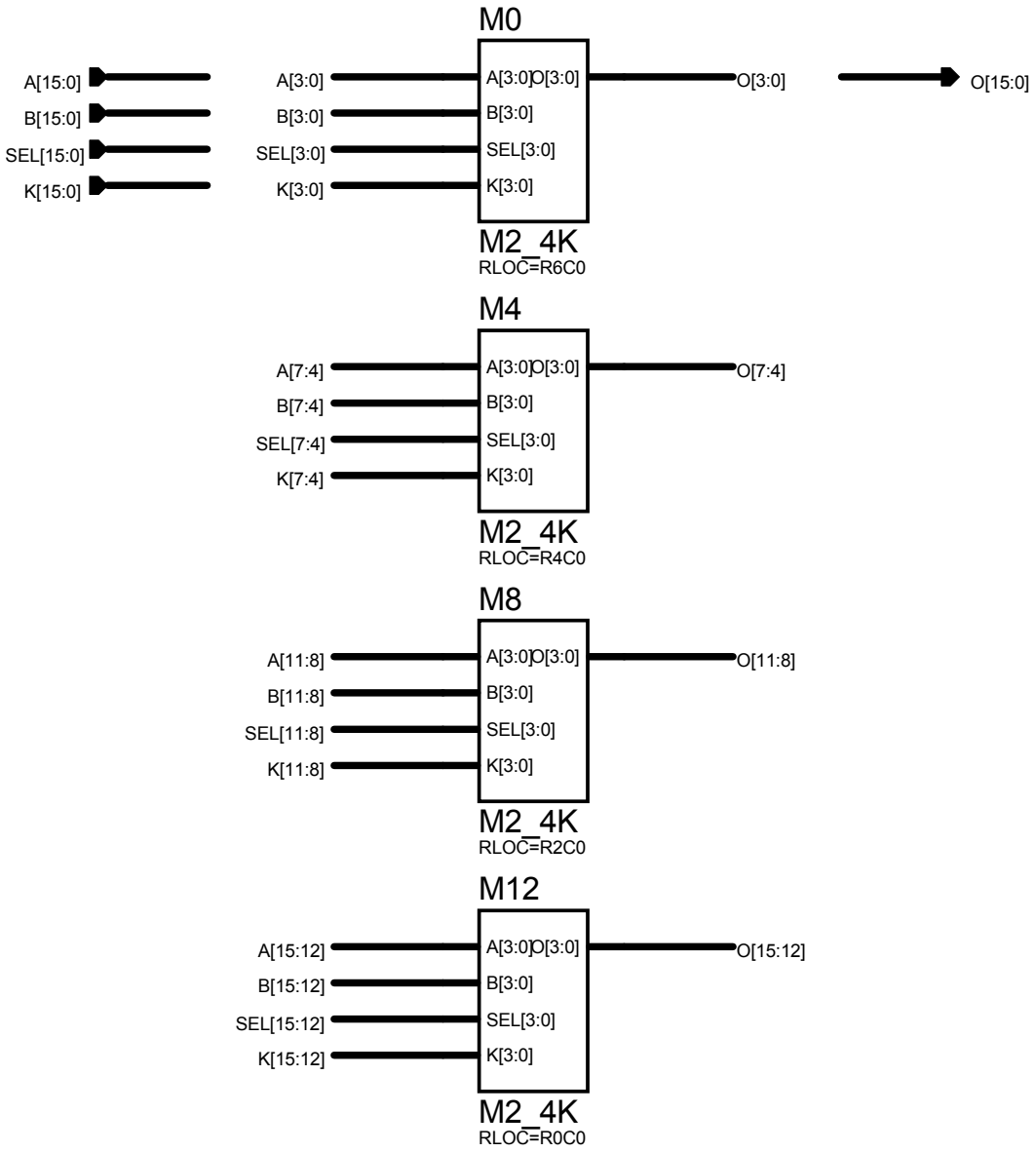




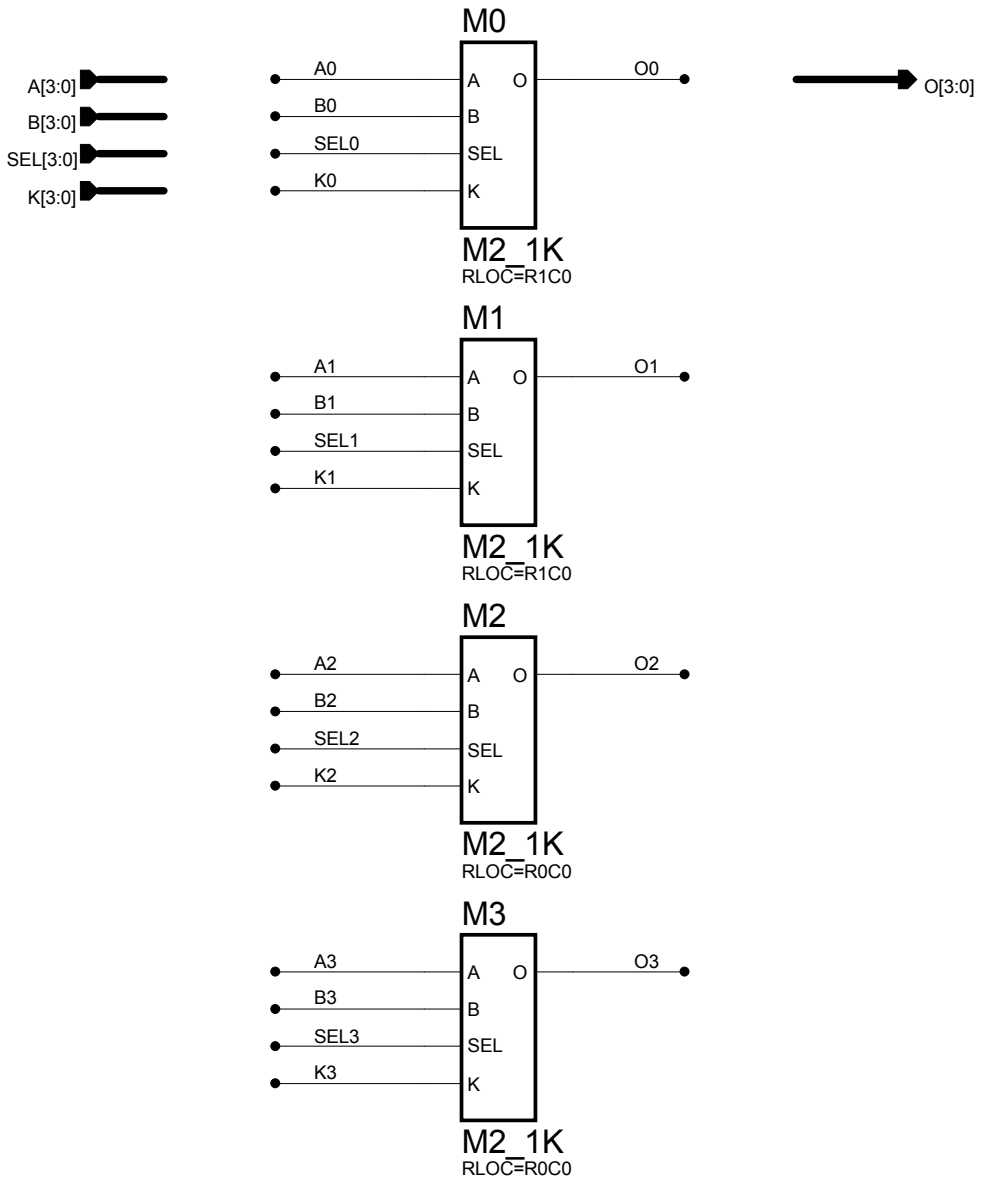
Project: [None]

Macro: M2_16Z

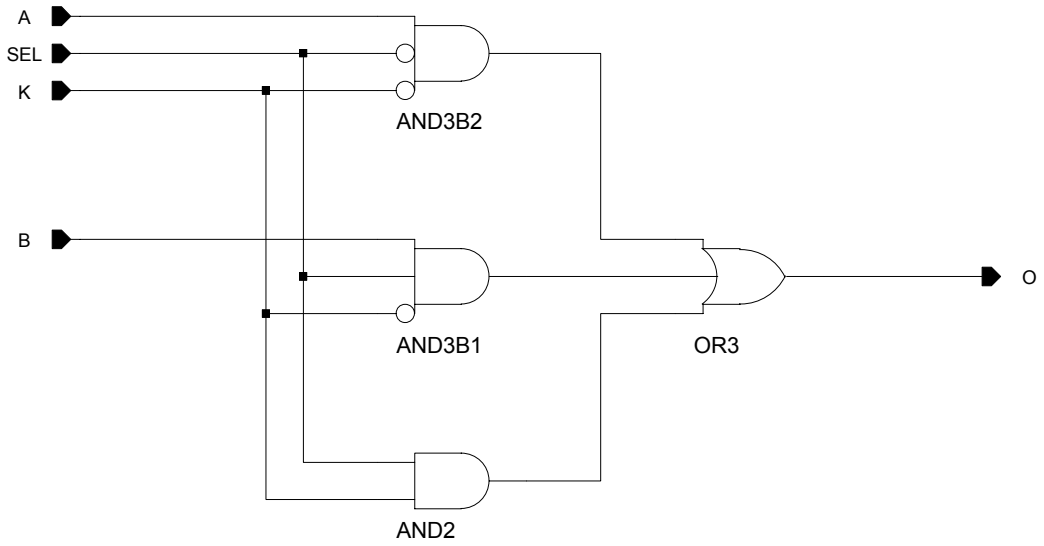
Date: 02/23/100



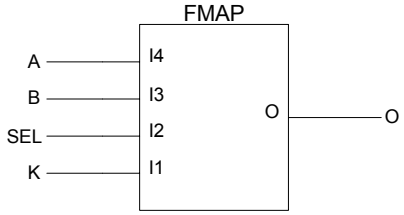
	Project: [None]
	Macro: M2_16K
	Date: 02/23/100



	Project: [None]
	Macro: M2_4K
	Date: 02/23/100

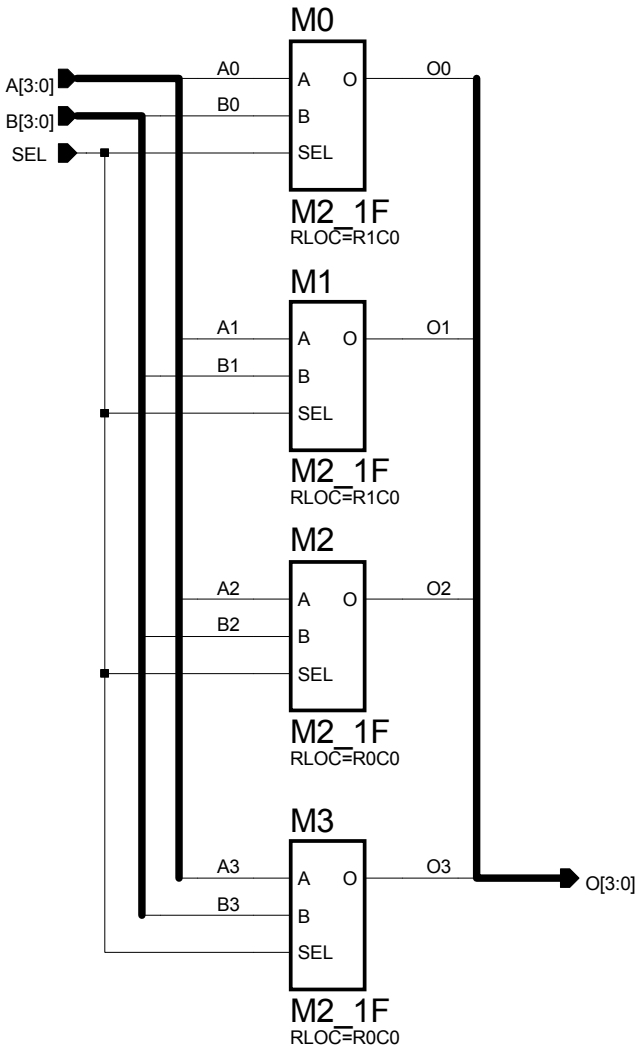


M0

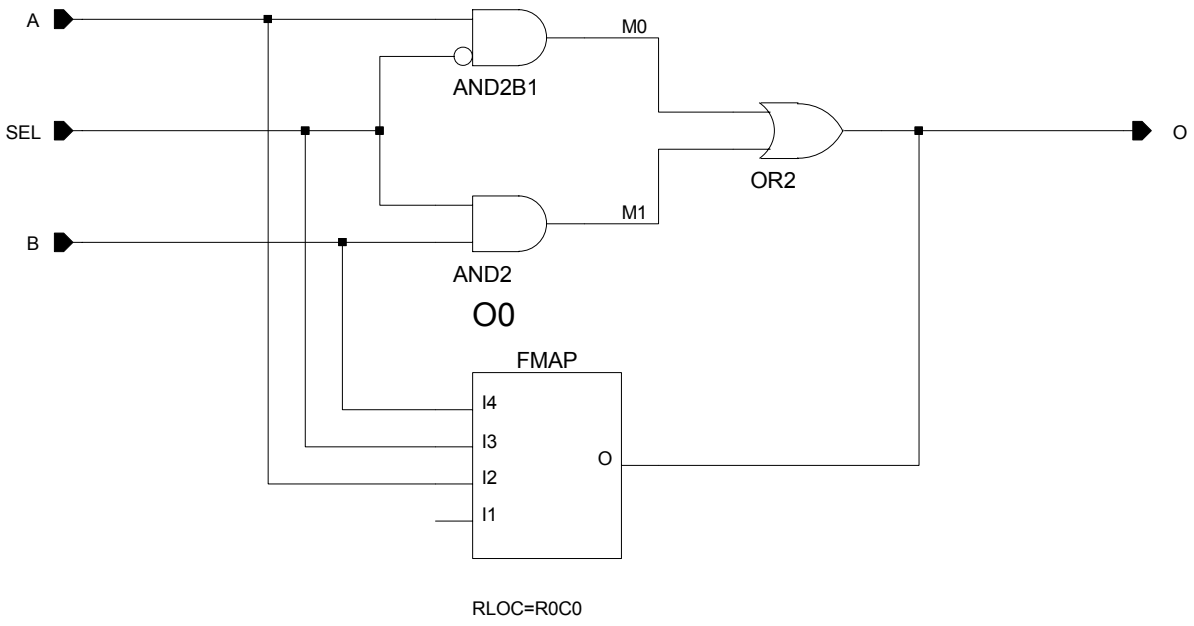


RLOC=R0C0

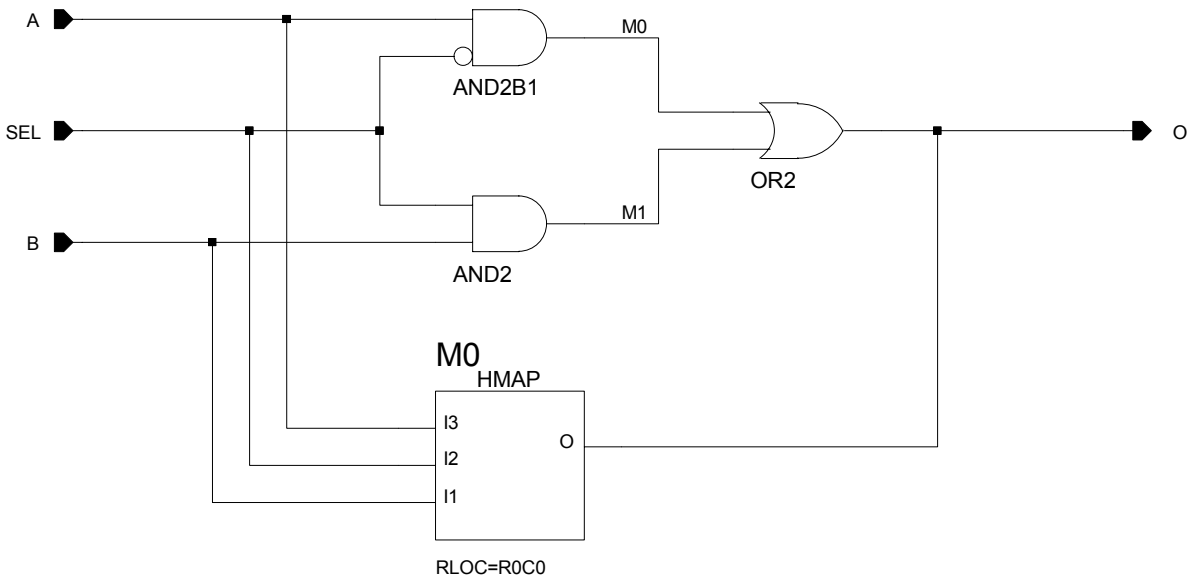
	Project: [None]
	Macro: M2_1K
	Date: 02/23/100



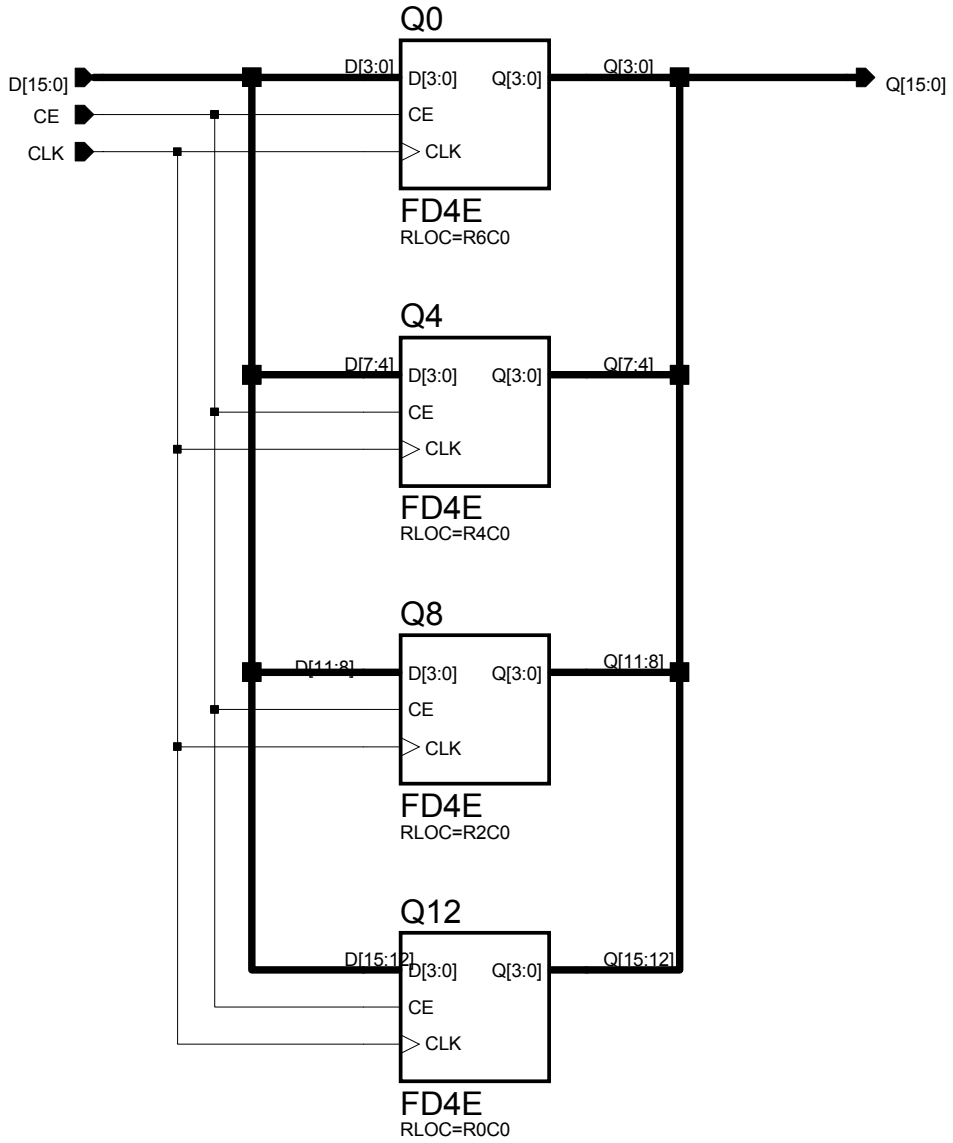
	Project: [None]
	Macro: M2_4
	Date: 02/23/100



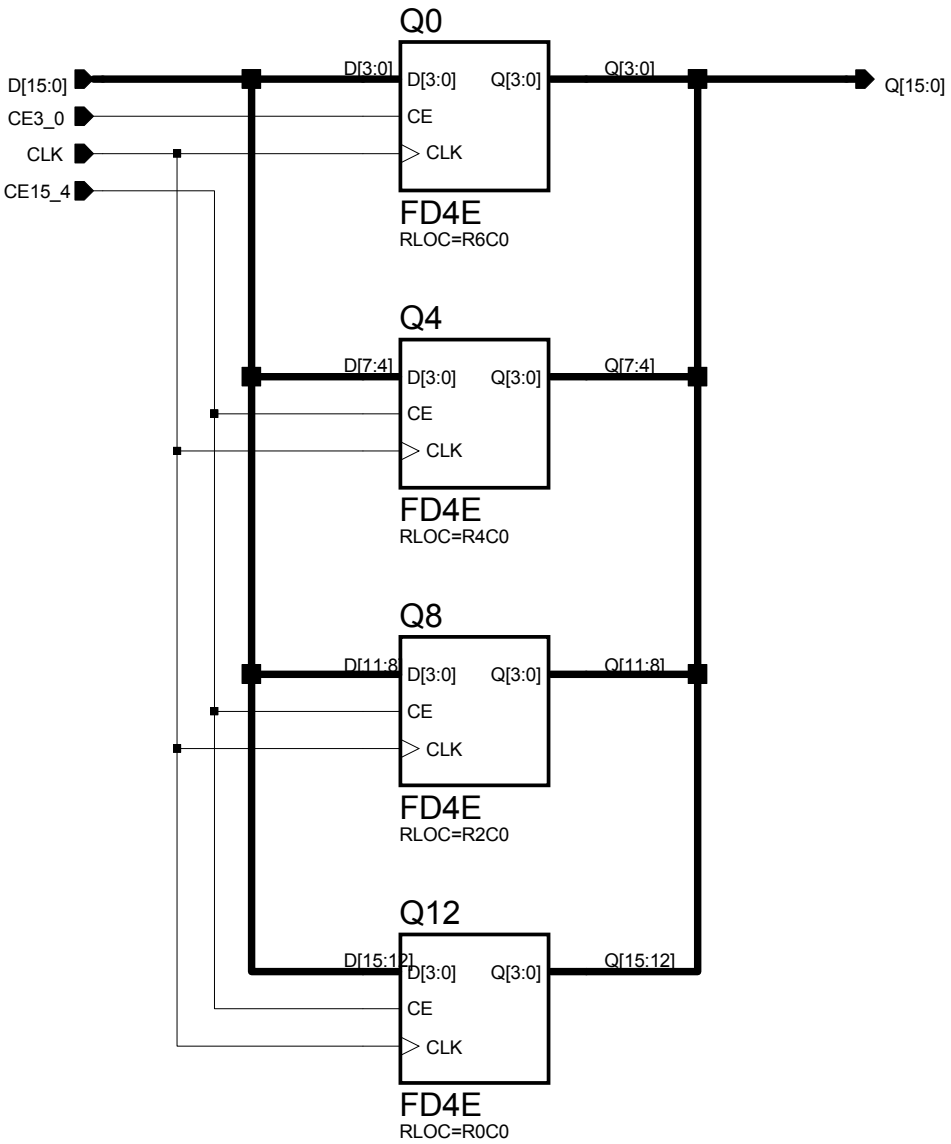
Project: [None]
Macro: M2_1F
Date: 02/23/100



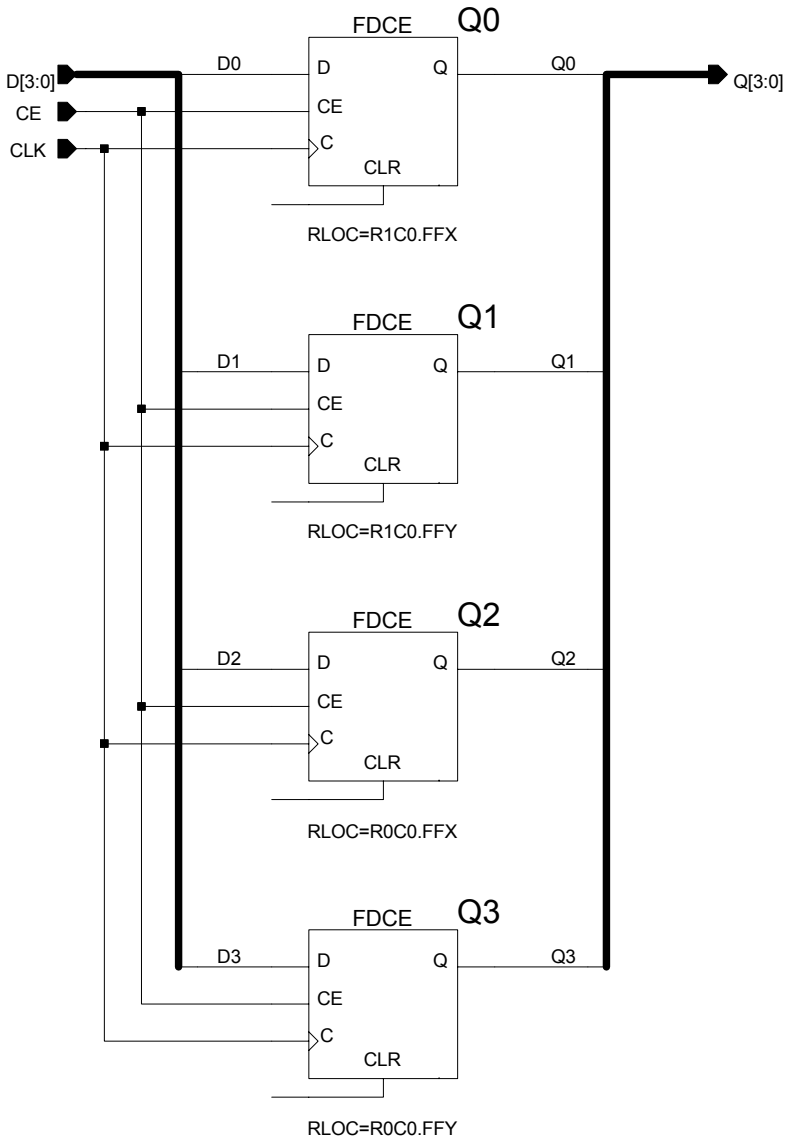
	Project: [None]
	Macro: M2_1H
	Date: 02/23/100



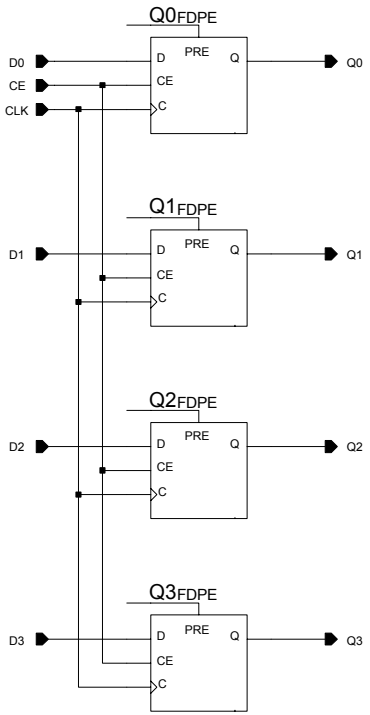
	Project: [None]
	Macro: FD16E
	Date: 02/23/100



Project: [None]
Macro: FD12E4E
Date: 02/23/100



	Project: [None]
	Macro: FD4E
	Date: 02/23/100



	Project: [None]
	Macro: FD4PE
	Date: 02/23/100