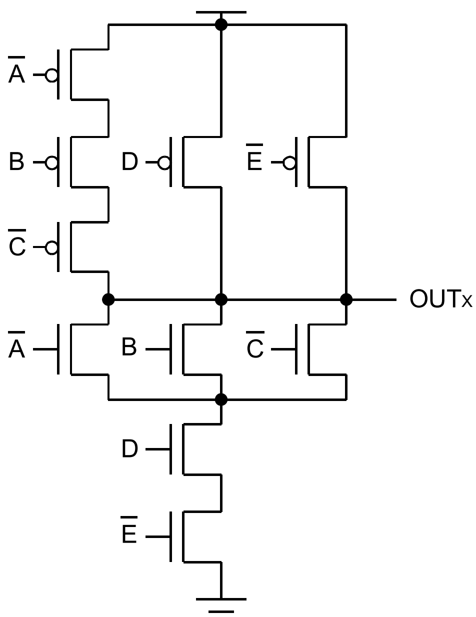
Problem 1 (4 parts, 32 points) Implementation Bonanza

For each part implement the specified device. **Label all inputs and outputs**.

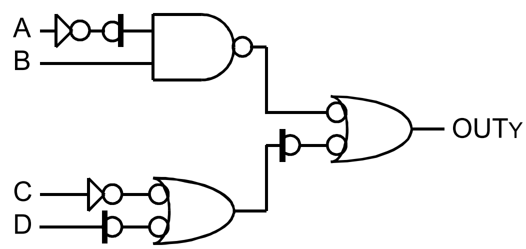
Part A (8 points) Implement the expression below using N and P type switches.

$$OUT_X = A \cdot \overline{B} \cdot C + \overline{D} + E$$

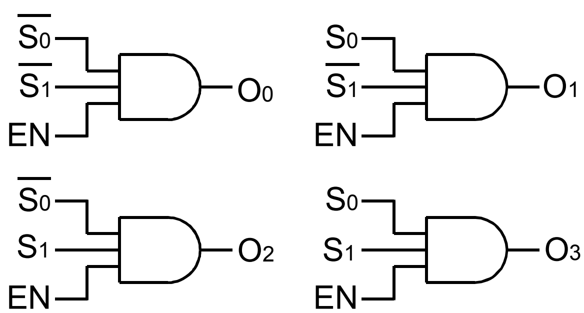


Part B (8 points) Implement the expression in mixed logic notation using NAND gates.

$$OUT_Y = \overline{A} \cdot B + (\overline{C + \overline{D}})$$



Part C (8 points) Implement a 2 to 4 decoder with enable using basic gates.



Part D (8 points) Write a POS expression for a two-input XOR (odd parity) using maxterms.

A	B	A XOR B
0	0	0
1	0	1
0	1	1
1	1	0

$$Out = (A + B) \cdot (\overline{A} + \overline{B})$$

Problem 2 (6 parts, 32 points) Alien Software

SETI has just received an interesting message from deep space. While the comments are written in an alien tongue, they appear to write programs in MIPS assembly. Intergalactic scientists have only been able to decode the register assignments. Computer engineers must take it from there.

```
# INPUTS: $1= num elements, $2= array A pointer, $3= array B pointer,
# OUTPUT: $6=result, WORKING: $4= InA/diff, $5= InB/pred,
```

#	label	instruction	comment
L1	WhatsIt:	sub $6, $6, $6	# clear max difference
L2	Loop:	lw $4, ($2)	# load A element
L3		lw $5, ($3)	# load B element
L4		sub $4, $4, $5	# compute difference
L5		slt $5, $4, $0	# if difference >= 0
L6		beq $5, $0, Skip1	# then skip
L7		sub $4, $0, $4	# otherwise negate
L8	Skip1:	slt $5, $6, $4	# if MaxDif >= difference
L9		beq $5, $0, Skip2	# then skip
L10		add $6, $4, $0	# otherwise update MaxDif
L11	Skip2:	addi $2, $2, 4	# increment A ptr
L12		addi $3, $3, 4	# increment B ptr
L13		addi $1, $1, -1	# decrement element count
L14		bne $1, $0, Loop	# loop if not done
L15		jr $31	# return to caller

Part A – E (26 points) Decode the abstract purpose of code in terms of the defined variable names. Don't transliterate instructions to words.

A: What does L1 accomplish?	B: What math function do L5-L7 implement?
It initializes Result to 0	absolute value
C: Why is Result updated (in terms of InA, InB)?	**D: What is the branch offset in L14 (in bytes)?**
|InA – InB| is greater than the current maximum difference.	-52 bytes (-13 instructions)

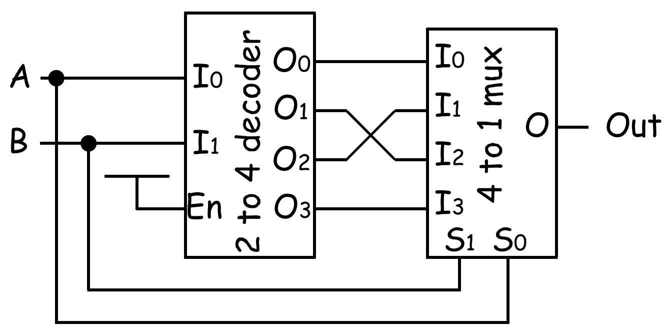
E: What does the overall function compute?
This function finds the largest difference in corresponding value in two equal sized arrays.

Part F (6 points) Another routine calls WhatsIt below. Add missing instructions to preserve and restore its return address on the stack. Recall that $29 is the stack pointer.

label	instruction	comment
	addi $29, $29, -4	# push return address …
	sw $31, ($29)	# … on stack
	jal WhatsIt	# call WhatsIt
	lw $31, ($29)	# pop return address …
	addi $29, $29, 4	# … off stack

Problem 3 (3 parts, 24 points) — Agents changed the matrix

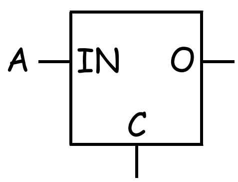
Part A (8 points) Consider the circuit below. Complete the truth table. Then state what logical function this circuit implements.



A	B	Out
0	0	1
1	0	0
0	1	0
1	1	1

This wacky circuit is a XNOR (even parity)

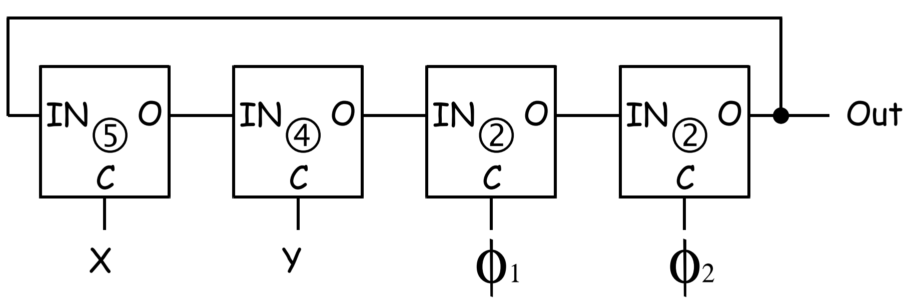
Part B (8 points) Consider four different function definitions below. The symbolic value A is presented at its input. The control input and resulting out are shown in the truth table. Name the *gate, building block, or storage device* that implements each definition.



In	C	②	③	④	⑤
A	0	Q_0	1	0	A
A	1	A		A	

② transparent latch ③ NAND ④ AND ⑤ XOR

Part C (8 points) Blocks from part B are used to create a new module below. The symbolic values X and Y are presented at its inputs along with a two-phase clock. Complete the truth table and give its functional name.



X	Y	Clk	Out
0	0	↑↓	0
1	0	↑↓	0
0	1	↑↓	Q_0
1	1	↑↓	

It's a: **toggle cell**

Problem 4 (2 parts, 28 points) Microcode

Using the supplied datapath, write microcode fragments to accomplish the following procedures. Express all values in hexadecimal notation. Use 'X' when a value is don't cared. For maximum credit, complete the description field.

Part A (14 points) $R_7 = \frac{3 \times R_5}{16} - 256 \times R_6$ **Modify only R_5, R_6 and R_7.**

#	X	Y	Z	rwe	im en	im va	au en	-a/s	lu en	lf	su en	st	ld en	st en	r/-w	msel	description
1	5	5	7	1	0	X	1	0	0	X	0	X	0	0	X	0	R7 □ R5 + R5
2	7	7	5	1	0	X	1	0	0	X	0	X	0	0	X	0	R7 □ R7 + R5
3	7	X	7	1	1	4	0	X	0	X	1	1	0	0	X	0	R7 □ R7 >> 4
4	6	X	6	1	1	FFF8	0	X	0	X	1	1	0	0	X	0	R6 □ R6 << 8
5	7	6	7	1	0	X	1	1	0	X	0	X	0	0	X	0	R7 □ R7 - R6
6																	
7																	

Part B (14 points) Write a microcode sequence that loads a 32 bit word from memory location 0x4000, unpacks and averages two 15 bit unsigned values (A and B), and then stores the result back to memory location 0x4000. Assume the most significant two bits of the register are zero. **Modify only R_1, R_2, and R_3.**

0	0	A	B
31	30	29 … 15	14 … 0

#	X	Y	Z	rwe	im en	im va	au en	-a/s	lu en	lf	su en	st	ld en	st en	r/-w	msel	description
1	X	X	1	1	1	4000	0	X	1	C	0	X	0	0	X	0	R1 □ 0x4000
2	1	X	2	1	0	X	0	X	0	X	0	X	1	0	1	1	R2 □ mem[R1]
3	2	X	3	1	1	7FFF	0	X	1	8	0	X	0	0	X	0	R3 □ R2 & 0x7FFF
4	2	X	2	1	1	F	0	X	0	X	1	0	0	0	X	0	R2 □ R2 >> 15
5	2	3	2	1	0	X	1	0	0	X	0	X	0	0	X	0	R2 □ R2 + R3
6	2	X	2	1	1	1	0	X	0	X	1	1	0	0	X	0	R2 □ R2 >> 1
7	1	2	X	0	0	X	0	X	0	X	0	X	0	1	0	1	mem[R1] □ R2
8																	

Problem 5 (4 parts, 40 points) This and That

Part A (9 points) Consider the instruction set architecture below with fields containing zeros.

0 0000	000 0000	000 0000	000 0000 0000 0000
opcode	dest. reg.	source 1 reg.	immediate value

What is the maximum number of opcodes? $2^5 = 32$

What is the number of registers? $2^7 = 128$

What is the range of the signed immediate value? $2^{15} = \pm 16K$

Part B (9 points) List three differences between a branch and a jump in the MIPS ISA.

1: Branches are conditional; jumps are unconditional.

2: Branch offsets are relative, jump targets are absolute.

3: Branch range ±32K I, ±128Kbytes; jump range 0-64M I, 0-256Mbytes

Part C (12 points) For 32 bit representations below, determine the most positive value and the step size (difference between sequential values). **All answers should be expressed in decimal notation**. Fractions (e.g., 3/16ths) may be used. Signed representations are two's complement.

representation	most positive value	step size
unsigned integer (32 bits) . (0 bits)	4 billion	1
signed fixed-point (28 bits) . (4 bits)	128 million	1/16
signed fixed-point (25 bits) . (7 bits)	16 million	1/128
signed fixed-point (21 bits) . (11 bits)	1 million	1/2K

Part D (10 points) Consider a memory system with **256 million addresses** of **8 byte words** using DRAM chips organized as **16 million** addresses by **32 bit words.**

word address lines for memory system: $\log_2(256M) = 28$

chips needed in one bank: 8 bytes / 4 bytes (32 bits) = 2 chips

banks for memory system: $256M / 16M = 2^8/2^4 = 2^4 = 16$ banks

memory decoder required (*n* to *m*): 4 to 16

total memory system size (in bytes): $256M \times 8 = 2^{28} \times 2^3 = 2^{31} = 2$GBytes