Exercices pour la révision de l’exame de mi-session

Problem 1. (24 points) A and B are 9-bit registers that contain the binary values: $A = 100011101$ and $B = 111110000$.

(a) (4 pts) A third 9-bit register R is used to store the result of the micro-operation $R \leftarrow A+B$. What will be the binary value in R as well as the two most significant output carries, c_8 and c_9, of the addition?

```
Carry:  11111100000
A:        100011101
B:      + 111110000
        -----------
R:        100001101
```

	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Unsigned Binary Nnumbers Converted to decimal	Decimal
A	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	= 256 + 16 + 8 + 4 + 1 = 285 =	**285**
B	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	= 256+128+64+32+16 = 496 =	**496**
R	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**1**	= 256 + 8 + 4 + 1 = **269 ≠**	**781**

(b) (4 pts) If the numbers in A and B are considered unsigned numbers, does R contain the correct result of $A + B$? Justify your answer.

No, R contains a wrong result (269_{10}), because the correct result 781 cannot be represented with 9 bits, which can represent positive numbers smaller than 512. Carry out **c_9** of the 9th bit of the A+B addition can be used to signalize the overflow of the two unsigned numbers A and B.

Signed 2's complement Representation

(c) (4 pts) If the numbers in A, B and B are considered signed numbers, represented in their 2's complement format, then does R contain the correct result of $A + B$? Justify your answer.

	sgn	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Magnitude	sgn	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Magnitude in decimal	Decimal
A	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	A<0 => |A| =-A=	0	1	1	1	0	0	0	1	1	=128+64+32+2 + 1 = 227 =	**-227**
B	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	B<0 => |B| =-B=	0	0	0	0	1	0	0	0	0	= 16 =	**-16**
R	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**1**	R<0 => |R|=-R=	0	1	1	1	1	0	0	1	1	=128+64+32+16+2+1= 243	**-243**

R contains the correct result (-243), since -243 ⊂ [-512,511], i.e., it can be correctly expressed with 9 bits in 2's complement representation. A circuit for signalizing overflow will agree:

(1) the carry out of 9th and 8th bits are the same or

(2) (2) the sign bit of the operands are the same with the result's sign bit (A_8=B_8=R_8=1)

2.

(a) (6 pts) If now the micro-operation $R \leftarrow A - B$ is executed, what will be the result stored in R as well as the last two output carries, c_8 and c_9, of the subtraction?

A – B = A + (2's complement of B) = A + B' + 1

```
Carry:   0000111111
A:         100011101
B':      + 000001111
         -----------
R:         100101101
```

(b) (6 pts) If the numbers in A and B are considered signed numbers, represented in their 2's complement format, does R contain the correct result of the subtraction? Justify your answer.

	c_9	c_8	c_7	c_6	c_5	c_4	c_3	c_2	c_1	c_0	Magnitude	sgn	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Magnitude in decimal	Decimal
	0	0	0	0	1	0	0	0	0	0												
A		**1**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	A<0 => |A| =-A=	0	1	1	1	0	0	0	1	1	=128+64+32+2 + 1 = 227 =	**-227**
-B		**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	-B>0 => |B| = B=	0	0	0	0	1	0	0	0	0	= 16 =	**16**
R		**1**	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**1**	R<0 => |R|=-R=	0	1	1	1	1	0	0	1	1	=128+64+32+16+2+1= 243	**-211**

Yes, R contains the correct result since -211 $\subset$ [-512,511] and, as such, it can be correctly represented in 2's complement format. There cannot be any overflow if adding a negative number to a positive number!

Circuits for overflow detection will asses that there is no overflow here since: the carry out of 9th and 8th bits are the same; or, by inspecting the sign bits of the operands and the result.

Problem 2. (42 points) In this problem, you will desgin a 3-bit ALU to perform the micro-operations described in Table 1. The ALU takes its operands from two 3-bit registers $A = A_2A_1A_0$ and $B = B_2B_1B_0$, and returns an output $F = F_2F_1F_0$. In the case of arithmetic operations, assume that the contents of A and B are signed numbers in 2's compelement representation.

Table 1: Function table of a 3-bit ALU.

Selection				
S_2	S_1	S_0	Operation	Description
0	0	0	$F = A - 1$	Decrement by 1 (Décrémentation par 1)
0	0	1	$F = A - B$	Subtruction (Soustraction)
0	1	0	$F = A - 2$	Decrement by 2 (Décrémentation par 2)
0	1	1	$F = A - B - 1$	Subtraction with borrow (Soustraction avec emprunt)
1	0	0	$F = \overline{A} \vee B$	Logic implication (Implication logique)
1	0	1	$F = A \oplus B$	Comparison (Comparaison)
1	1	0	$F = cir\ A$	Circular shift right (Décalage circulaire à droite)
1	1	1	$F = ashl\ A$	Arithmetic shift left (Décalage arithmétique à gauche)

1. Design of arithmetic unit

(a) (10 pts) Draw a detailed logic circuit of the ALU's arithmetic unit.

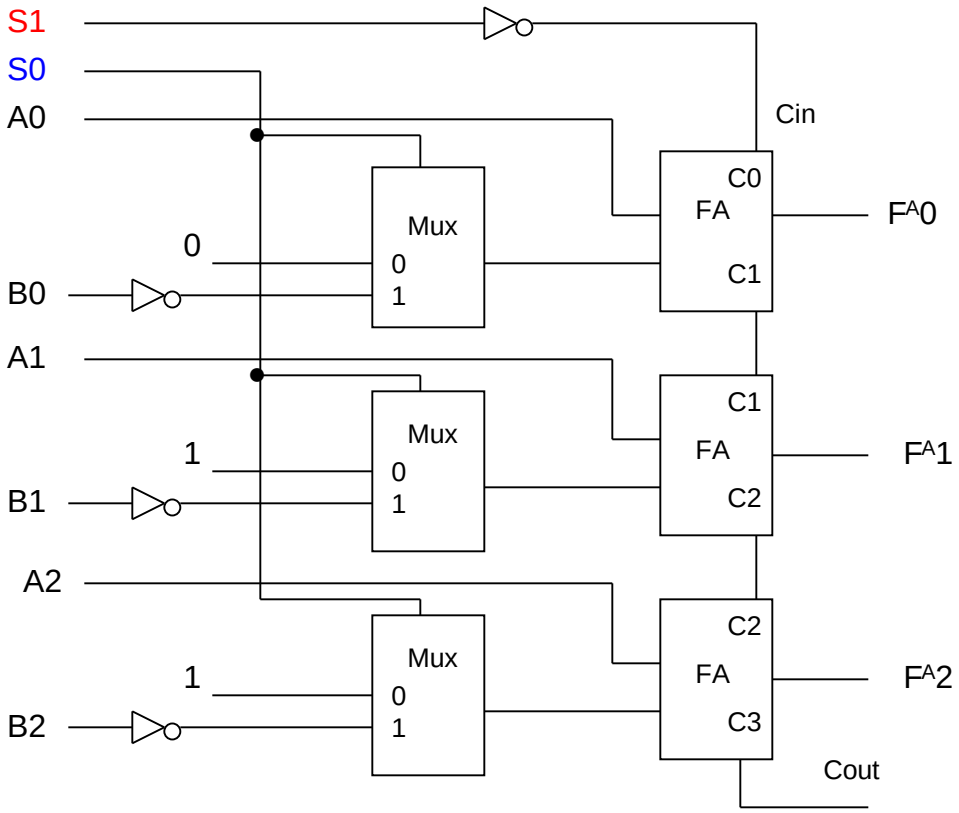
S_2	S_1	S_0	Operation	Description
0	0	0	$F = A - 1$	Decrement by 1 (Décrémentation par 1)
0	0	1	$F = A - B$	Subtruction (Soustraction)
0	1	0	$F = A - 2$	Decrement by 2 (Décrémentation par 2)
0	1	1	$F = A - B - 1$	Subtraction with borrow (Soustraction avec

F^0=A-1=A+2's(001) = A+111=(A+110)+1
F^1=A-B=A+2's(B2 B1 B0)=A+(B2' B1' B0'+1)

F^2=A-2=A+2's(010) = (A+110)
F^3=A-B-1=A+2's(B2 B1 B0)-1=A+(B2' B1' B0'+1)-1
=A+(B2' B1' B0')

F^0=F^2+1 => $F^{0,2}$ = (A+110)+S1'
F^1=F^3+1 => $F^{1,3}$ = (A+B2'B1'B0')+S1'

F^A = S0' $F^{0,2}$ + S0 $F^{1,3}$ =
= S0' [(A+110)+S1'] + S0 [(A+B2'B1'B0')+S1']



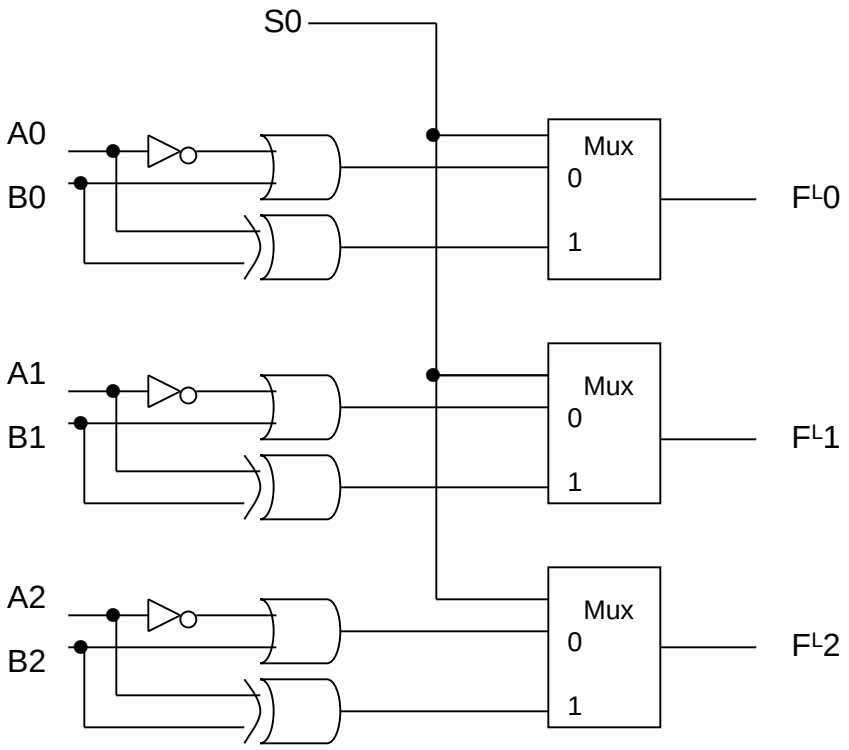
(b) (3 pts) A Boolean variable W is set to 1 when an overflow occurs and is reset to 0 otherwise. Find a simplified Boolean expression of W.

$W = C3 \oplus C2$

2. Design of logic unit

Draw a detailed logic circuit of the ALU's logic unit.

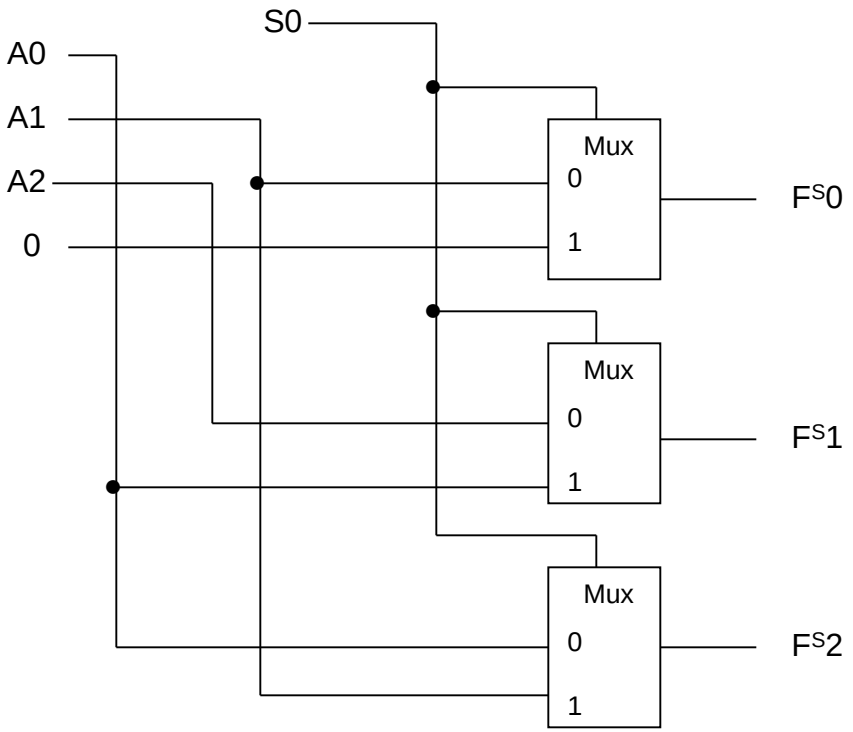
1	0	0	$F = \overline{A} \vee B$	Logic implication (Implication logique)
1	0	1	$F = A \oplus B$	Comparison (Comparaison)



3. Design of shift unit

(a) (8 pts) Draw a detailed logic circuit of the ALU's shift unit.

1	1	0	$F = cir\ A$	Circular shift right (Décalage circulaire à droite)
1	1	1	$F = ashl\ A$	Arithmetic shift left (Décalage arithmétique à gauche)



Present state $A_2 A_1 A_0$

cir A : $A_0 A_2 A_1$

Present state $A_2 A_1 A_0$

ashl A : $A_1 A_0 0$

(b) (3 pts) A Boolean variable V is set to 1 when an overflow occurs during the arithmetic shift and is reset to 0 otherwise. Find a simplified Boolean expression of V.

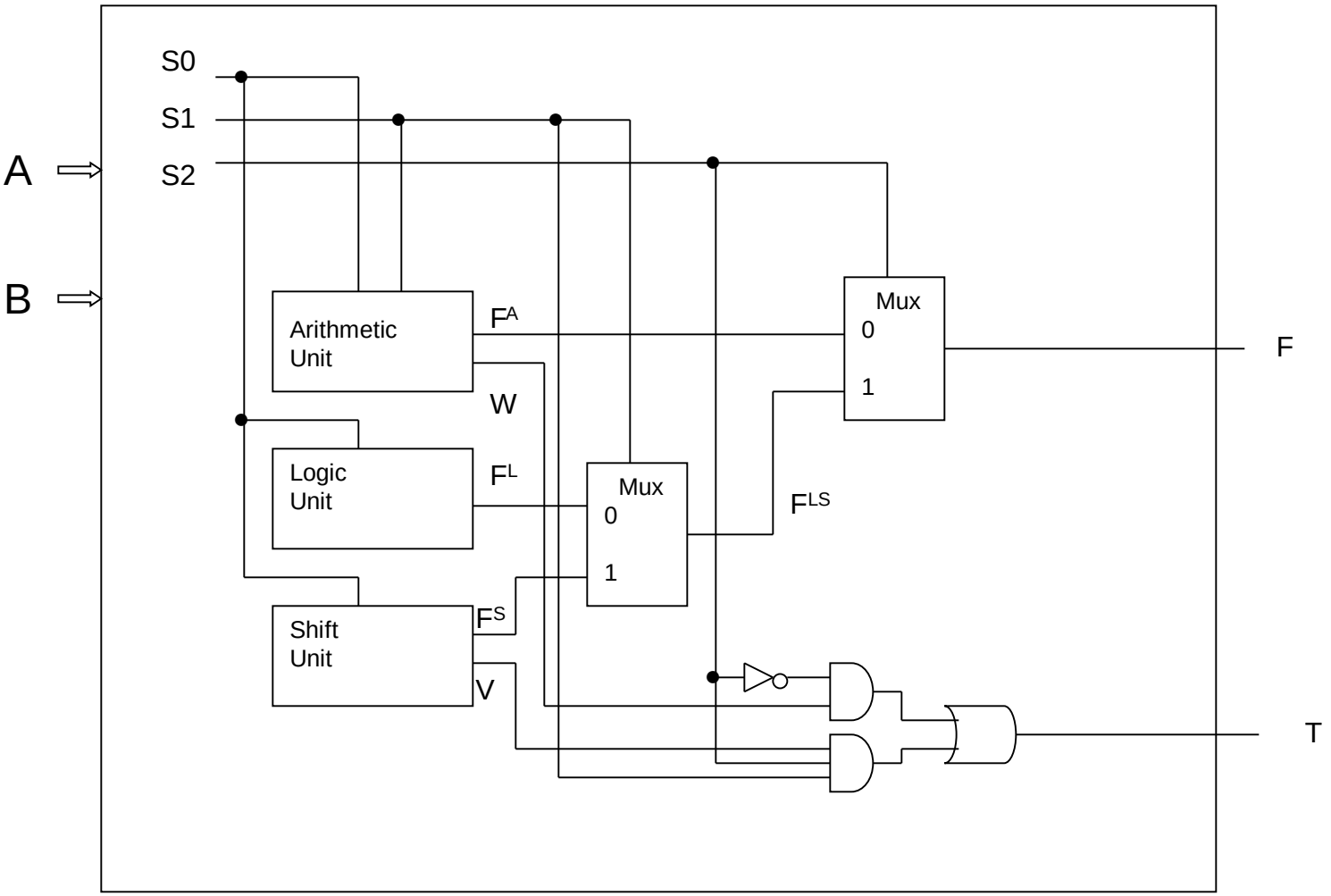
$V = S0 (A2 \oplus A1)$

4. Finalizing the ALU design

(a) (4 pts) A Boolean variable T is used to determine if one the micro-operations stated in Table 1 caused an overflow. T is set to 1 when an overflow occurs and is reset to 0 otherwise. Find a simplified Boolean expression of T. (*Hint:* express T in terms of W and V, and possibly other Boolean variables).

$T = W S2' + V S2 S1$

(b) (6 pts) Use bloc diagrams of the arithmetic, logic and shifting units in order to draw the bloc diagram of the complete ALU, including the overflow detection bit T.



Probleme 3. (34 points) Figure 1 shows the state diagram of a logic circuit which has a unique one-bit external input x.

1. Start by deriving the state table of the circuit. Then, assuming that JK flip-flops are to be used in the implementation, extend the state table with the excitation table of the circuit.

Present State$^{(n)}$		Input	Next State$^{(n+1)}$		Flip flop inputs			
Q_1	Q_0	x	Q_1	Q_0	J_1	K_1	J_0	K_0
0	0	0	0	0	0	d	0	d
0	0	1	0	1	0	d	1	d
0	1	0	0	0	0	d	d	1
0	1	1	1	1	1	d	d	0
1	0	0	d	d	d	d	d	d
1	0	1	d	d	d	d	d	d
1	1	0	0	1	d	1	d	0
1	1	1	1	1	d	0	d	0

Q_1 \ $Q_0 x$	00	01	11	10
0	0	0	1	0
1	d	d	d	d

$J_1 = Q_0 x$

Q_1 \ $Q_0 x$	00	01	11	10
0	d	d	d	d
1	d	d	0	1

$K_1 = \overline{x}$

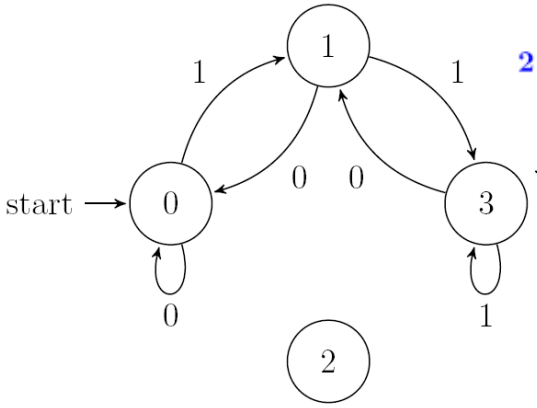


Figure 1: State diagram

2. Find simplified expressions for each flip-flop inputs.

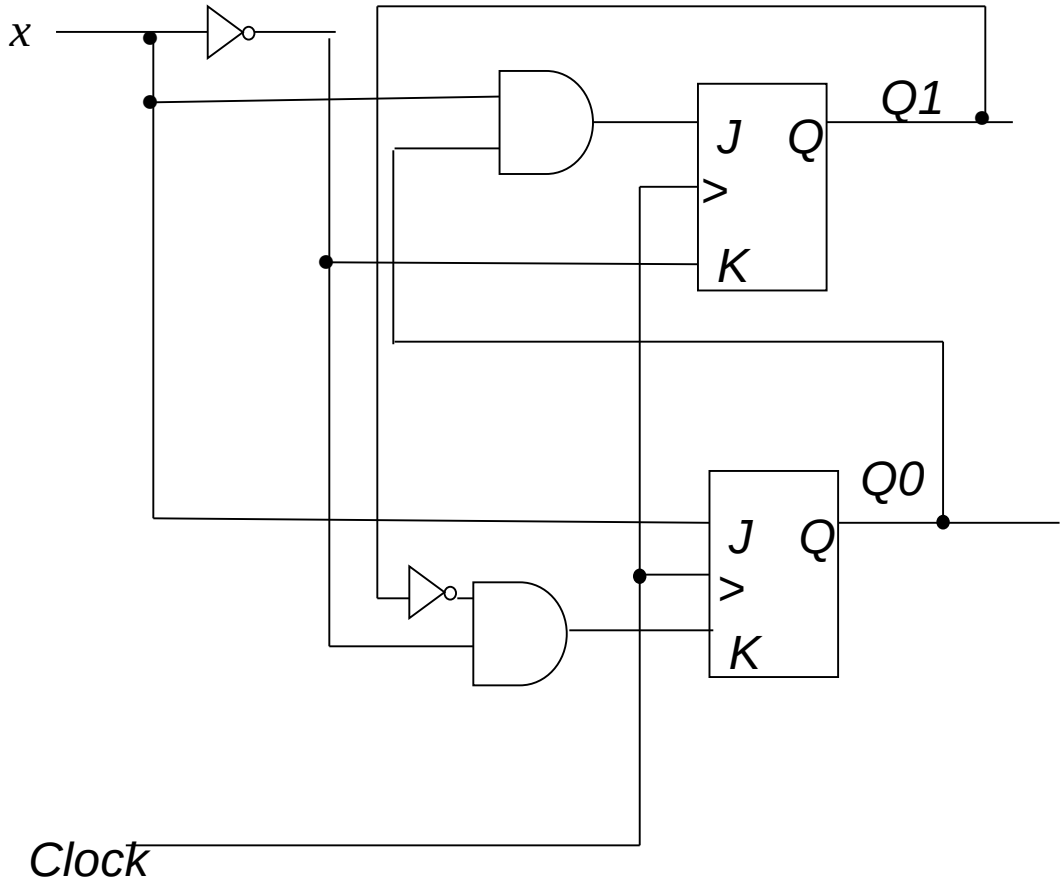
Q_1 \ $Q_0 x$	00	01	11	10
0	0	1	d	d
1	d	d	d	d

$J_0 = x$

Q_1 \ $Q_0 x$	00	01	11	10
0	d	d	0	1
1	d	d	0	0

$K_0 = \overline{Q_1}\,\overline{x}$

3. Draw the logic circuit using JK flip-flops and the minimum number of simple logic gates (AND/OR/NOT)



3-16. Perform the arithmetic operations (+42) + (-13) and (-42) - (-13) in binary using signed-2's complement representation.

+42 = 0101010
−42 = 1010110

(+42) 0101010
(−13) 1110011
(+29) 0011101

+13 = 0001101
−13 = 1110011

(−42) 1010110
(+13) 0001101
(−29) 1100011

3-17. Perform the arithmetic operations (+70) + (+80) and (-70) + (-80) with binary numbers in signed-2' s complement representation. Use eight bits to accommodate each number together with its sign. Show that overflow occurs in both cases, that the last two carries are unequal, and that there is a sign reversal.

3-17

0 1 ← last two carries → 1 0

+70 01000110
+80 01010000
+150 10010110

greater than +127

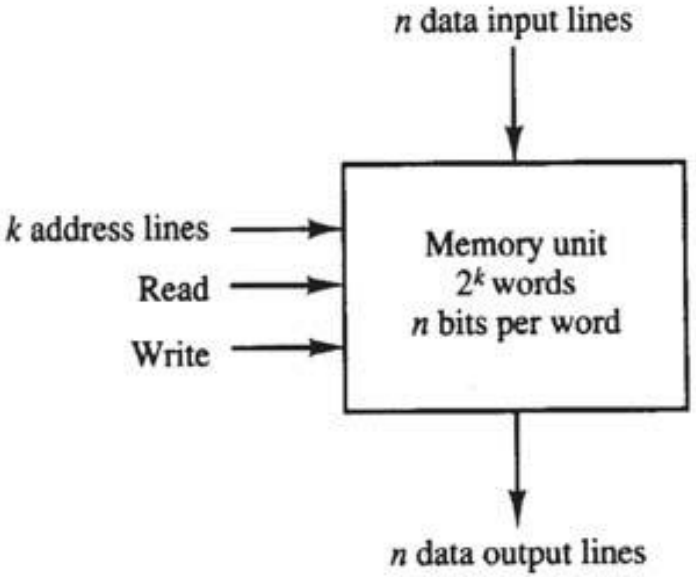
negative

-70 10111010
-80 10110000
-150 01101010

less than −128

positive

- The following memory units are specified by the number of words times the number of bits per word. How many address lines and input-output data lines are needed in each case?



	Address lines	Data lines
(a) $2K \times 16 = 2^{11} \times 16$	11	16
(b) $64K \times 8 = 2^{16} \times 8$	16	8
(c) $16M \times 32 = 2^{24} \times 32$	24	32
(d) $4G \times 64 = 2^{32} \times 64$	32	64

- Specify the number of bytes that can be stored in the memories listed above

(a) $2K \times 2 = 4K = 4096$ bytes

(b) $64K \times 1 = 64K = 2^{16}$ bytes

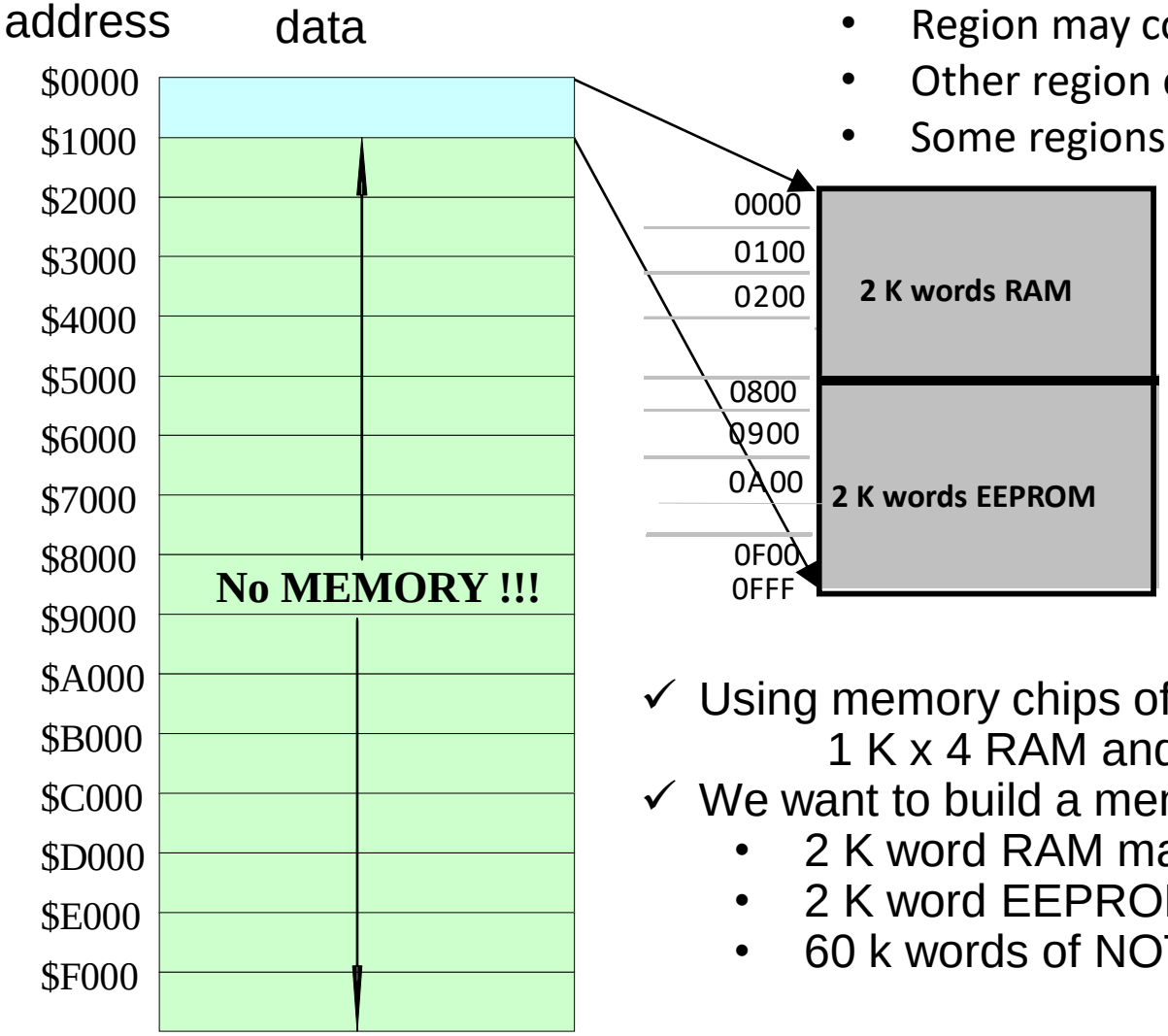
(c) $2^{24} \times 4 = 2^{26}$ bytes

(d) $2^{32} \times 8 = 2^{35}$ bytes

Memory Map

Memory **map** shows how all memory addresses are used:

- Region may contain RAM
- Other region contains ROM
- Some regions contain nothing



Mano's **Basic Computer** has

- ✓ Words of 16 **data** bits
- ✓ 12 bits for Address, so its memory space is 4-kwords;
  - ✓ since its CPU has 16 bit bus, it may have provisions for further expanding the memory to 64 k-words.

- ✓ Using memory chips of
  1 K x 4 RAM and 1 K x 8 EEPROM
- ✓ We want to build a memory as follows:
  - 2 K word RAM mapped to $0000 - $07FF
  - 2 K word EEPROM ($0800 to $0FFF)
  - 60 k words of NOTHING ($1000 to $FFFF)

ADDR	ADDR (HEX)	$A_{15}A_{14}A_{13}A_{12}$	$A_{11}A_{10}A_9A_8$	$A_7A_6A_5A_4$	$A_3A_2A_1A_0$	$D_{15}D_{14}D_{13}D_{12}$	$D_{11}D_{10}D_9D_8$	$D_7D_6D_5D_4$	$D_3D_2D_1D_0$
0000	0000	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0				
0001	0001	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	RAM 0,3	RAM 0,2	RAM 0,1	RAM 0,0
…	…	…	…	…	…	1K X 4	1K X 4	1K X 4	1K X 4
1023	03FF	0 0 0 0	0 0 1 1	1 1 1 1	1 1 1 1				
1024	0400	0 0 0 0	0 1 0 0	0 0 0 0	0 0 0 0				
1025	0401	0 0 0 0	0 1 0 0	0 0 0 0	0 0 0 1	RAM 1,3	RAM 1,2	RAM 1,1	RAM 1,0
…	…	…	…	…	…	1K X 4	1K X 4	1K X 4	1K X 4
2047	07FF	0 0 0 0	0 1 1 1	1 1 1 1	1 1 1 1				
2048	0800	0 0 0 0	1 0 0 0	0 0 0 0	0 0 0 0				
2049	0801	0 0 0 0	1 0 0 0	0 0 0 0	0 0 0 1				
…	…	…	…	…	…				
0BFF	0BFF	0 0 0 0	1 0 1 1	1 1 1 1	1 1 1 1	EEPROM 1		EEPROM 0	
	0C00	0 0 0 0	1 1 0 0	0 0 0 0	0 0 0 0	2K X 8		2K X 8	
	0C01	0 0 0 0	1 1 0 0	0 0 0 0	0 0 0 1				
	…								
4095	0FFF	0 0 0 0	1 1 1 1	1 1 1 1	1 1 1 1				

Basic Computer Memory