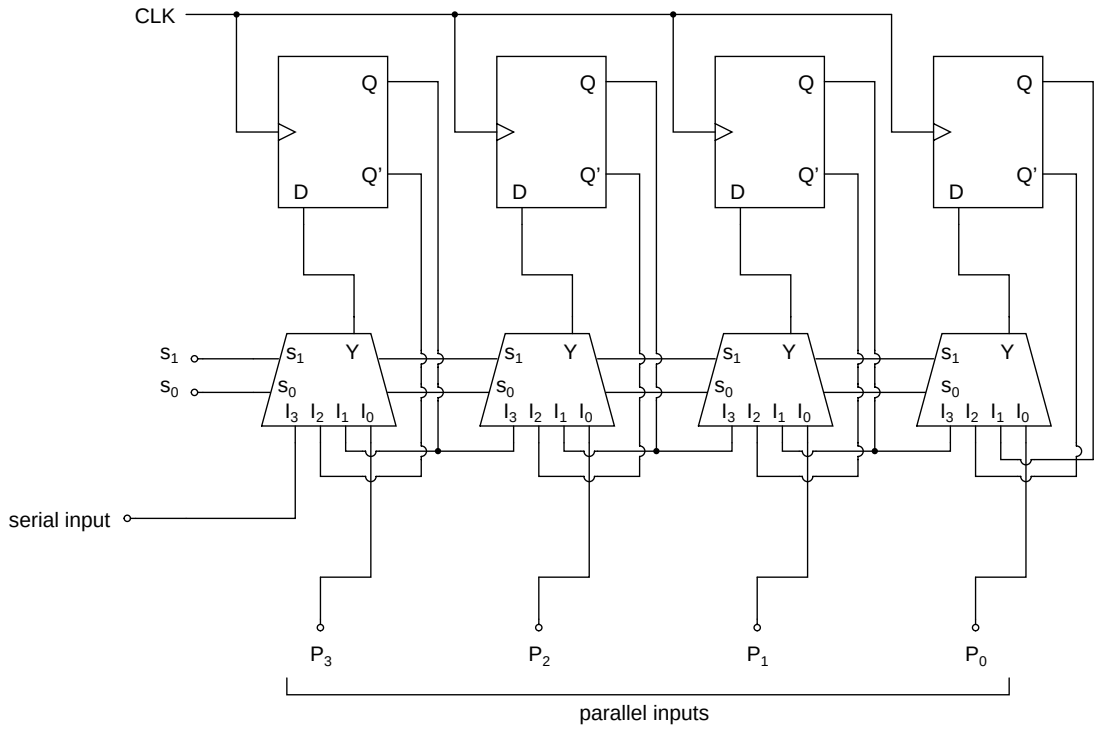
INEL 4205 - Extra Practice Problems Solutions - Exam 4 – Spring 2009

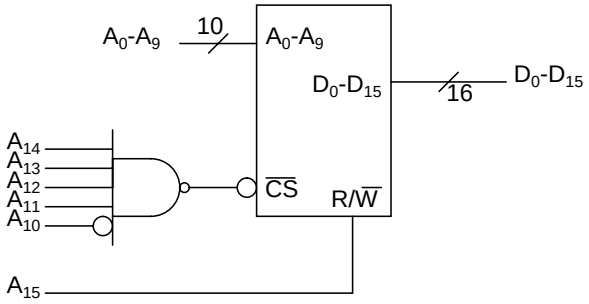
This problems are just examples and are not indicative of the exam that you will take.
You will prepare better for the exam by first reading the chapters 6 and 7 and working on the textbook problems.

1. The following diagram shows four D-flip flops and four 4×1 multiplexers. The multiplexer inputs, output and select terminals are indicated by the letters I, Y, and s, respectively. Notice that the select lines s1s0 are common to all multiplexers. Complete the diagram so that the circuit operates as follows:

$s_1 s_0$	action
00	Parallel load – loads parallel inputs into register
01	Hold – keeps the register contents
10	Negate – complements the register contents
11	Shift right; serial input is inserted in first (left) flip flop



2. The following diagram shows how a RAM chip is connected to a CPU that have 16-bit data and address buses.



a) How many bytes of information can be stored in the RAM? Answer: 2048 bytes.

b) Determine the addresses that the CPU can use to read data from RAM. For each range, express the beginning and ending address in hex.

Answer: Five address bits are used to select the chip. One address bit is used to select read or write. Lower bound is $1111100000000000 = F800$. Upper bound is $1111101111111111 = FBFF$.

Range is from $F800$ to $FBFF$.

c) Repeat part b, but for the addresses that the CPU can use to write data into RAM.

Answer: Lower bound is $0111100000000000 = 7800$. Upper bound is $0111101111111111 = 7BFF$.

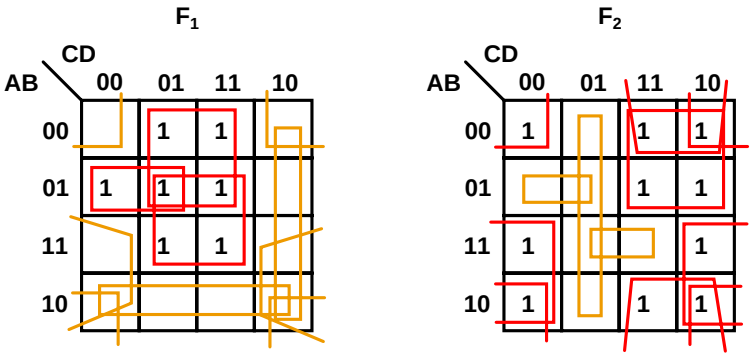
Range is from 7800 to $7BFF$.

3. Determine the PLA programming table needed to implement the following two boolean functions. Minimize the number of product terms. Show all your work, including the Karnaugh maps used in the minimization.

$$F1(A,B,C,D) = \sum(1,3,4,5,7,13,15)$$
$$F2(A,B,C,D) = \sum(0,2,3,6,7,8,10,11,12,14)$$

Answer:

The following K-maps



produce the following expressions:

$$F_1 = A'BC' + A'D + BD \quad (1)$$
$$= A'BC' + A'D + ABD \quad (2)$$
$$= (AB' + CD' + B'D' + AD')' \quad (3)$$
$$F_2 = B'D' + A'C + B'C + AD' \quad (4)$$
$$= (A'BC' + ABD + C'D)' \quad (5)$$

Equation 2 is obtained by forming a group of two 1's instead of the group of four 1's used to obtain equation 1. To find the combination with least distinct terms, we compare the expressions. The result can be summarized as follows:

F_1	F_2	distinct terms
1	4	7
2	4	7
3	4	6
1	5	5
2	5	4
3	5	7

So we select equation 2 for F_1 and equation 5 for F_2, which results in the following table.

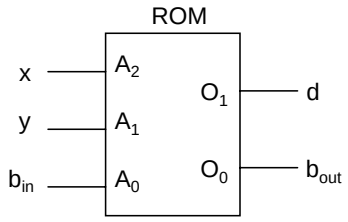
Terminos	Entradas A	B	C	D	Salidas ( ) F_1	( ) F_2
A'BC'	0	1	0	-	1	1
A'D	0	-	-	1	1	-
ABD	1	1	-	1	1	1
C'D	-	-	0	1	-	1

Nota: es posible que no necesite usar algunas filas.

4. Show how to use a ROM to implement a full-subtractor that finds the difference x-y between the two 1-bit inputs x and y, and accepts a borrow input bin. The output should be a 1-bit difference d and a 1-bit borrow bout. Fill the following table with the contents of the ROM. Specify how do you connect the variables to the chip's inputs and outputs on the schematic diagram, shown on the right-hand-side.

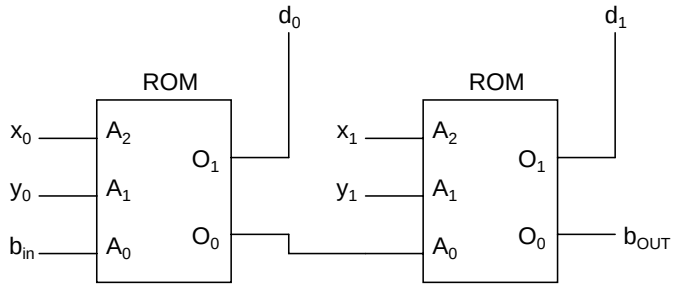
   Answer: This is one possible way of solving the problem.

Address A_2	A_1	A_0	Content O_1	O_0
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

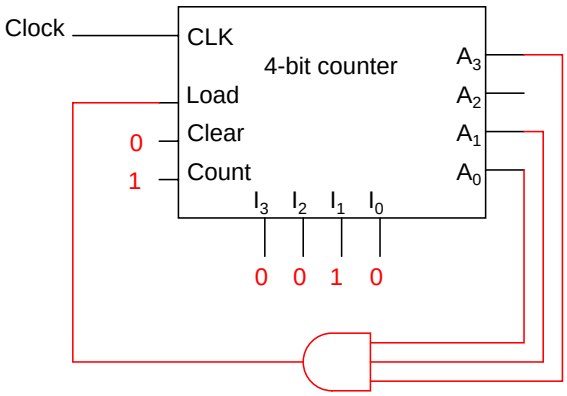


5. On the diagram shown below, specify and label the external connections that you would use to construct a 2-bit subtractor using two ROMs like the one developed in problem 4. The new circuit should accept two 2-bit numbers x1x0 and y1y0 and a borrow input bin. The output should be a 2-bit difference d1d0 and a 1-bit borrow bout. You should write on the diagram where would the input and output bits are to be connected and also how to connect the first ROM borrow output to the second ROM.

   Answer: The following diagram shows how to connect the chips if you use the solution shown in problem 4.



6. The following sketch represents a 4-bit binary counter with parallel load and clear inputs. Applying a logic-1 in the "Load" input causes the signals in inputs $I_3I_2I_1I_0$ to be loaded into the counter. A logic-1 in the "Clear"input resets the count to 0000. A logic-1 at the "Count" input enables counting; a logic-0 at this input prevents counting. The device count appears in the outputs $A_3A_2A_1A_0$. Show how to construct a device that will count from 2 to 11 repetitively using this counter, by connecting inputs to logic-1, logic-0 or by adding external logic gates. Initially, your circuit might go through counts 0000 and 0001, but only the first time power is applied to the device.



7. The stimulus program shown below is used to simulate the binary counter with parallel load described in the module named "counter". Going over the program, predict what would be the output of the counter and the carry output from t=0 to t=155ns.

```
//Binary counter with parallel load
module counter (Count,Load,IN,CLK,Clr,A,CO);
input Count,Load,CLK,Clr;
input [3:0] IN; //Data input
output CO; //Output carry
output [3:0] A; //Data output
reg [3:0] A;
assign CO = Count & ~Load & (A == 4'b1111);
always @ (posedge CLK or negedge Clr)
if (~Clr) A = 4'b0000;
else if (Load) A = IN;
else if (Count) A = A + 1'b1;
else A = A; // no change, default condition
endmodule
// Stimulus for testing counter
module testcounter;
```

```
reg Count, Load, CLK, Clr;
reg [3:0] IN ;
wire C0 ;
wire [3:0] A ;
counter cnt (Count, Load, IN, CLK, Clr, A, C0)
always
#5 CLK = ~CLK ;
initial
begin
Clr = 0;
CLK = 1;
Load = 0; Count = 1;
#5 Clr = 1;
#50 Load = 1; IN = 4'b1100;
#10 Load = 0;
#70 Count = 0;
#20 $finish;
end
endmodule
```

Answer:

After $5ns$ the *Count* signal becomes active, and until $50ns$ the count increments every $10ns$ on the positive edge of the clock. The count starts from 0 and it is 5 at $50ns$. The *Load* signal then becomes active at $55ns$ and changes the count to 12 on the positive edge of the clock at $60ns$. *Load* becomes inactive at $65ns$ and the counter increments the count to 13 on the following positive edge of the clock at $70ns$. After that the count increments by 1 every $10ns$ until the *Count* signal goes low at $135ns$, stopping the count at 3.

8. A 12-bit Hamming code word containing 8 bits of data and 4 parity bits is read from memory. What was the original 8-bit data word that was written into memory if the 12-bit word read out is as follows:

   a) 101110000110
   Answer:

$$\begin{aligned} C_1 &= XOR(b_1, b_3, b_5, b_7, b_9, b_{11}) = XOR(1,1,1,0,0,1) = 0 \\ C_2 &= XOR(b_2, b_3, b_6, b_7, b_{10}, b_{11}) = XOR(0,1,0,0,1,1) = 1 \\ C_4 &= XOR(b_4, b_5, b_6, b_7, b_{12}) = XOR(1,1,0,0,0) = 0 \\ C_8 &= XOR(b_8, b_9, b_{10}, b_{11}, b_{12}) = XOR(0,0,1,1,0) = 0 \end{aligned}$$

   Bit number $0010_2 = 2_{10}$ is in error. The correct Hamming code word is 111110000110. The data, which is obtained from the Hamming code word after removing the parity bits at positions 1, 2, 4 and 8, is $\boxed{11000110}$.

   b) 101111110100
   Answer:

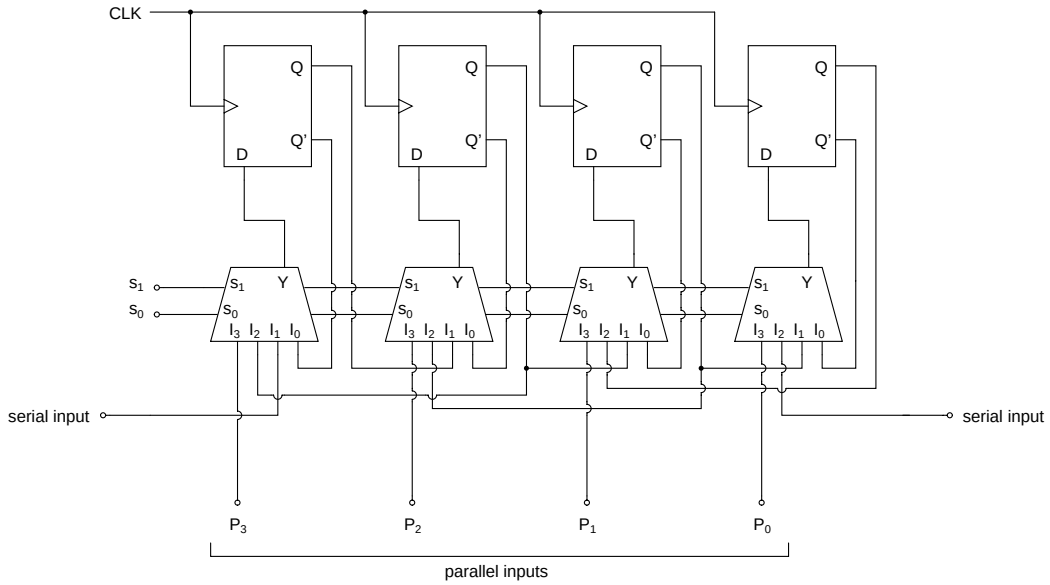
$$\begin{aligned} C_1 &= XOR(b_1, b_3, b_5, b_7, b_9, b_{11}) = XOR(1,1,1,1,0,0) = 0 \\ C_2 &= XOR(b_2, b_3, b_6, b_7, b_{10}, b_{11}) = XOR(0,1,1,1,1,0) = 0 \\ C_4 &= XOR(b_4, b_5, b_6, b_7, b_{12}) = XOR(1,1,1,1,0) = 0 \\ C_8 &= XOR(b_9, b_{10}, b_{11}, b_{12}) = XOR(1,0,1,0,0) = 0 \end{aligned}$$

   There are no errors. The data, which is obtained from the Hamming code word after removing the parity bits at positions 1, 2, 4 and 8, is $\boxed{11110100}$.

9. Draw the logic diagram of a 4-bit register with four D flip-flops and four 4×1 multiplexers with mode selection inputs s_1 and s_0. The register operates according to the following function table.

$s_1 s_0$	Register Operation
00	Complement the four outputs
01	Shift right
10	Shift left
11	Load parallel data

Answer:



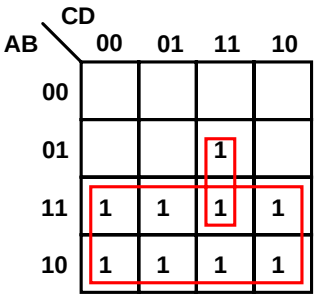
10. Mark the fuse map of the PAL, using the following sketch and indicating a connection with an X, to indicate how it should be programmed to implement the following Boolean functions, given in sum of minterms form:

$$
\begin{aligned}
F_1(A,B,C,D) &= \sum(7,8,9,10,11,12,13,14,15) \\
F_2(A,B,C,D) &= \sum(2,12,13) \\
F_3(A,B,C,D) &= \sum(0,2,3,4,5,6,7,8,10,11,15) \\
F_4(A,B,C,D) &= \sum(1,2,8,12,13)
\end{aligned}
$$

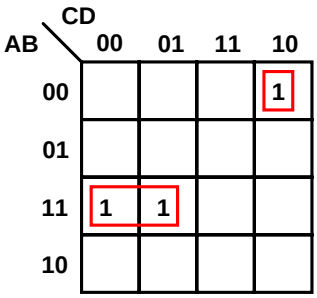
Appropriately label the inputs and outputs by writing the variable name inside the corresponding square. Show any other work in the next page.

Answer:

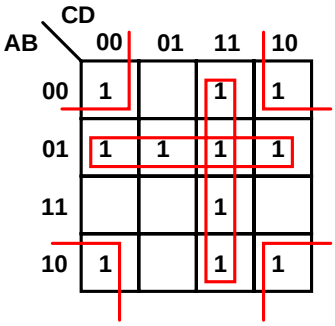
F_1



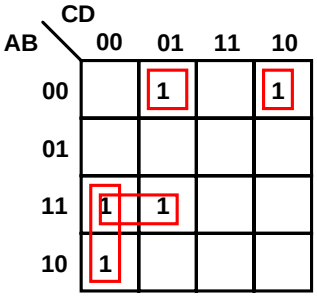
F_2



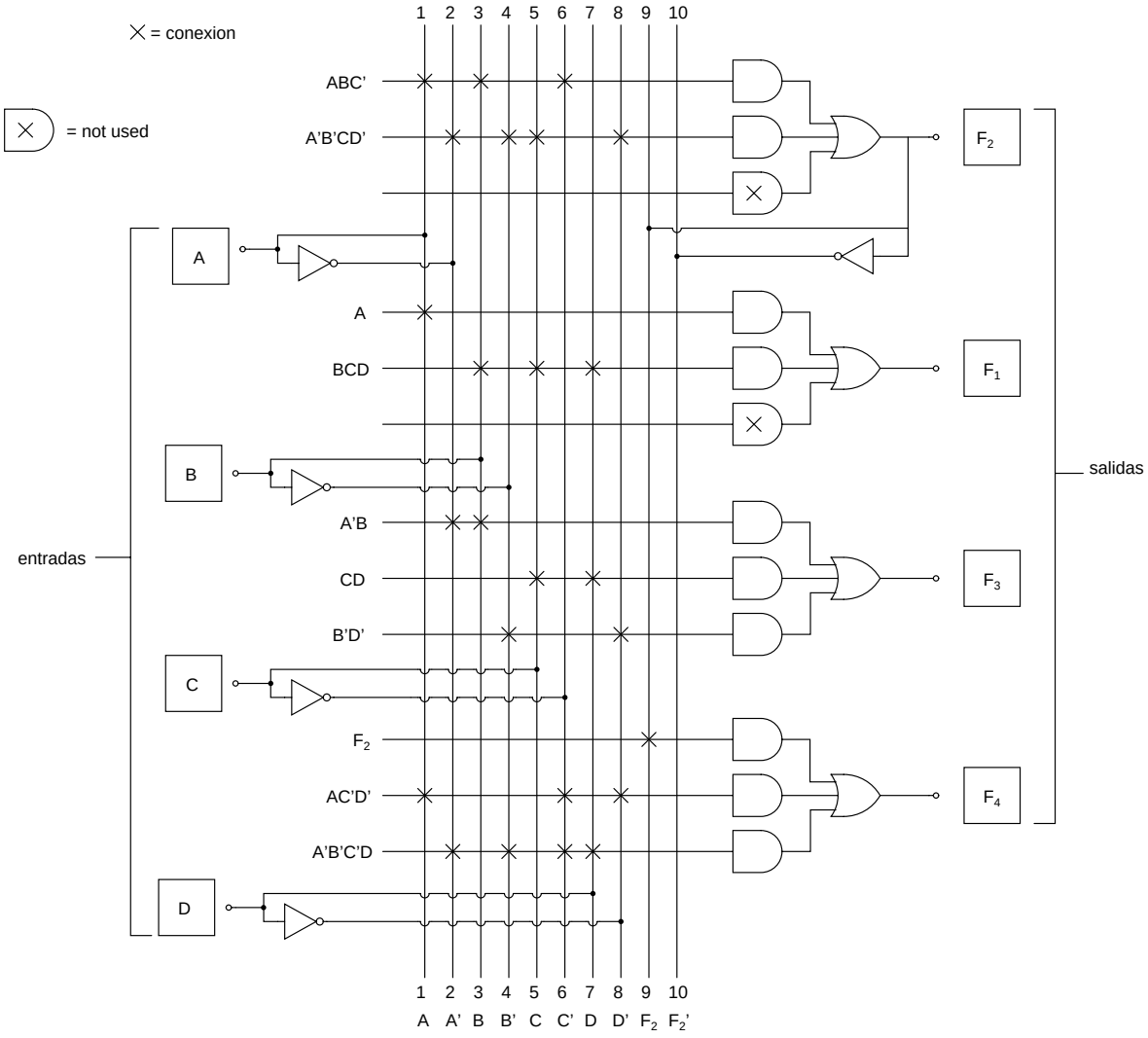
F_3



F_4



$$
\begin{aligned}
F_1 &= A + BCD \\
F_2 &= ABC' + A'B'CD' \\
F_3 &= A'B + CD + B'D' \\
F_4 &= AC'D' + A'B'C'D + ABC' + A'B'CD' = AC'D' + A'B'C'D + F_2
\end{aligned}
$$


× = conexion
= not used
1 2 3 4 5 6 7 8 9 10
ABC'
A'B'CD'
A
BCD
B
A'B
CD
B'D'
C
F2
AC'D'
A'B'C'D
D
entradas
F2
F1
F3
F4
salidas
1 2 3 4 5 6 7 8 9 10
A A' B B' C C' D D' F2 F2'