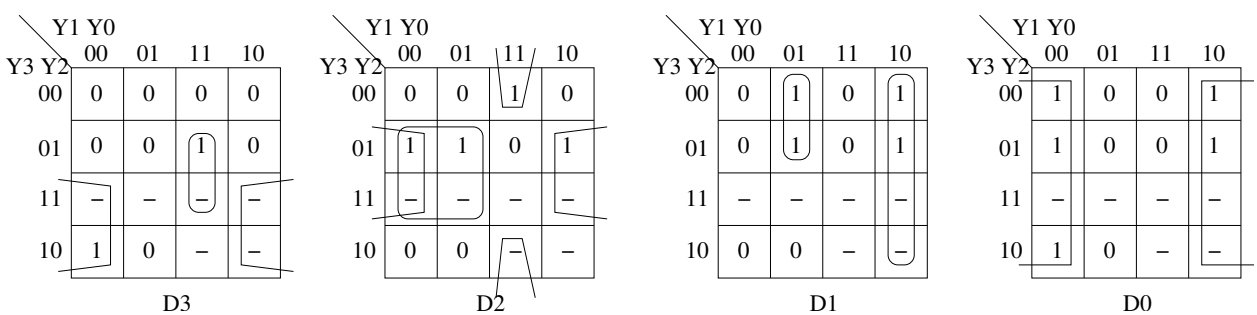
CIS 251-101 Computer Organization	**Homework 5** **Solution**	Dr. David Nassimi Fall 2004

1. Design a 4-bit synchronous BCD counter (Y_3, Y_2, Y_1, Y_0) with D flip-flops by applying formal design procedures. The BCD counter has consecutive counts $0, 1, \ldots, 9$ and then back to 0. The counter has **no external inputs** except the CLOCK, and it advances its count every clock cycle.

Solution:

Pres.-State				Next-State			
Y_3	Y_2	Y_1	Y_0	Y_3^n	Y_2^n	Y_1^n	Y_0^n
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0

For D flip-flops, $D_i = Y_i^n$. We form the map for each flip-flop input and obtain the expression.



$D_3 = Y_3\overline{Y_0} + Y_2Y_1Y_0$

$D_2 = Y_2\overline{Y_1} + Y_2\overline{Y_0} + \overline{Y_2}\ Y_1Y_0 = Y_2 \oplus (Y_1Y_0)$

$D_1 = Y_1\overline{Y_0} + \overline{Y_3}\ \overline{Y_1}\ Y_0$

$D_0 = \overline{Y_0}$

2. A 3-bit register $Y = (Y_2, Y_1, Y_0)$ has two control inputs (S_1, S_0) which specify the operation to be performed as follows:

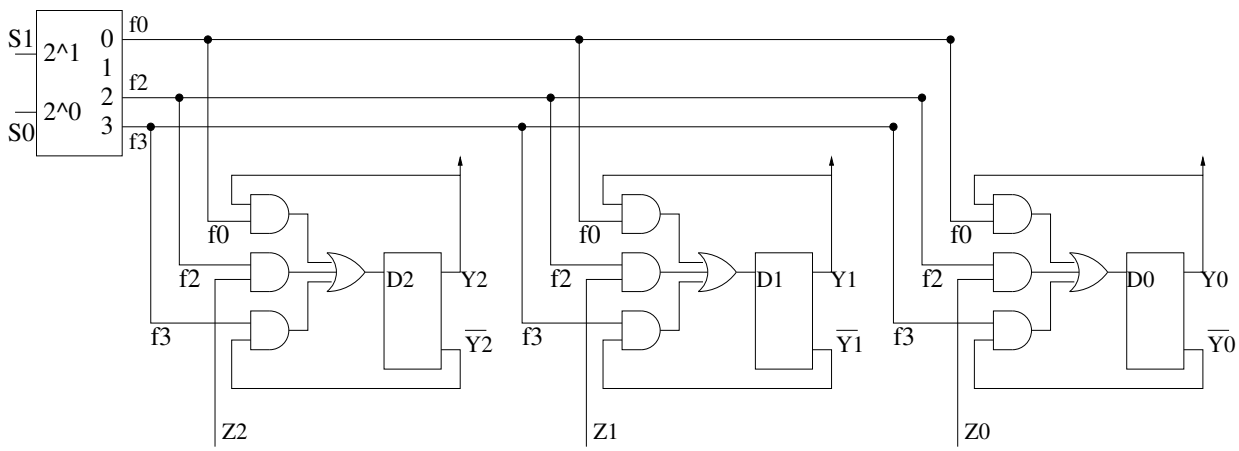
S_1	S_0	Register Operation
0	0	No change
0	1	Clear the register Y
1	0	Parallel Load from input $Z = (Z_2, Z_1, Z_0)$
1	1	Complement the register

Design this register using D flip flops. First apply formal procedures to find the logic for the least-significant bit of the register. (That is, find the Boolean expression for D_0, which is the the input for the least-significant-bit.) Then show the complete circuit for the 3-bit register. You may find it convenient to use a 2-to-4 decoder for the control inputs (S_1, S_0).

Solution:

S_1	S_0	D_0
0	0	Y_0
0	1	0
1	0	Z_0
1	1	$\overline{Y_0}$

$$\begin{aligned} D_0 &= \overline{S_1}\ \overline{S_0}\ Y_0 + S_1\overline{S_0}\ Z_0 + S_1 S_0 \overline{Y_0} \\ &= f_0 Y_0 + f_2 Z_0 + f_3 \overline{Y_0} \qquad \text{(The decoder has outputs } f_0, f_1, f_2, f_3.\text{)} \end{aligned}$$



3. Consider two registers Y and Z. There are three control inputs C_1, C_2, C_3, each enabling a transfer operation as follows. (For example, when $C_2 = 1$, data is copies from Z to Y.)

Control Input	Operation	Explanation
C_1 :	$Y \leftarrow 0$	Clear register Y
C_2 :	$Y \leftarrow Z$	Register transfer from Z to Y
C_3 :	$Y \leftarrow \overline{Z}$	Transfer $\overline{Z}$ to Y

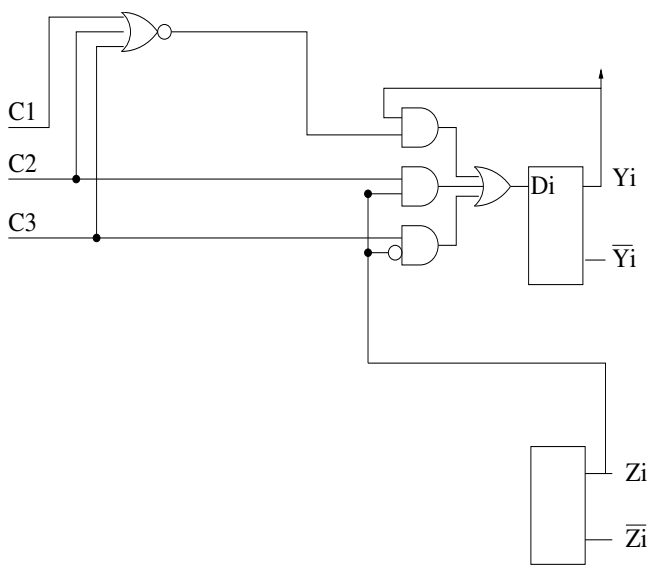
The control variables are mutually exclusive. That is, only one of control inputs can be 1 at any time, while the other two are 0. (If all three control inputs are 0, no action takes place.)

Draw one bit-slice, say bit i, of the logic diagram. Use D-type flip-flops.

Solution:

This problem is very similar to Problem 2 with some minor differences: The control signals are not encoded (thus no need for a decoder), and the parallel input now comes from register Z. Let D_i be the input to Y_i flip-flop. The expression for D_i may be written easily:

$$D_i = Y_i^n = (\overline{C_1 + C_2 + C_3}\ )Y_i + C_2 Z_i + C_3 \overline{Z_i}$$



4. A 4-bit ALU (Arithmetic Logic Unit), which is entirely combinational circuit, has two 4-bit inputs A and B and a 4-bit output Z. The function performed by the ALU is determined by two control inputs (S_1, S_0) as follows:

S_1	S_0	Z
0	0	$A+B$
0	1	$A-B$
1	0	A
1	1	$A+1$

Design the ALU using Full-Adders and other logic gates. Formally derive the logic for one bit slise of the ALU, and the logic for the carry into the least significant bit. Then, draw the complete circuit.

Hint: Derive the logic for one bit slice as follows. Let X_i and Y_i be the inputs to the Full-Adder for bit slice i. For each function of the ALU, figure out X_i and Y_i in terms of A_i and B_i. Similary, figure out C_0, the carry into the least-significant bit.

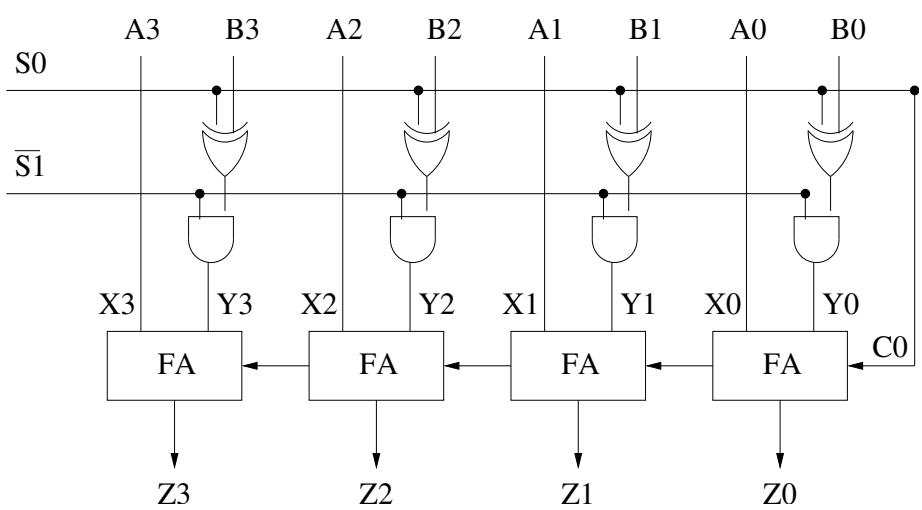
You may use a truth table to tabulate these. (List S_1 and S_0 as explicit variables, so the table will have four rows, corresponding to the four functions of the ALU.)

Solution:

Let X_i and Y_i be the inputs to the full-adder for bit slice i, FA_i. And let C_0 be the input-carry for the least-significant bit FA_0. We will use a truth table to determine X_i, Y_i and C_0.

S_1	S_0	X_i	Y_i	C_0
0	0	A_i	B_i	0
0	1	A_i	$\overline{B_i}$	1
1	0	A_i	0	0
1	1	A_i	0	1

$$X_i = A_i$$

$$Y_i = \overline{S_1}\ \overline{S_0}\ B_i + \overline{S_1}\ S_0\overline{B_i} = \overline{S_1}\ (S_0 \oplus B_i)$$

$$C_0 = \overline{S_1}\ S_0 + S_1 S_0 = S_0.$$