Midterm 1 Cheat Sheet

1 Chapter 1

This chapter is mostly a review of ITI 1100.

2 Chapter 2

2.1 Multifunction Circuits

This is a circuit that preforms more than one function. For example, a circuit can store if $S = 0$, and count up if $S = 1$.

A1A0	Function
00	Store
01	Count up
10	Load
11	Clear

Next we will make a state table with the following columns (assuming we are doing a D flip flop which is easy and 4 bits)

A1A0	Q3Q2Q1Q0	Q3'Q2'Q1'Q0'	D3D2D1D0
operation	*current state*	*next state*	*flip flop inputs*

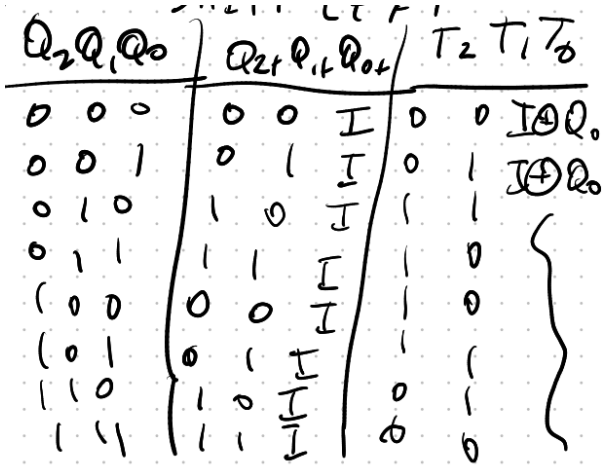
Then we can create all 4 flip flops, and then put a MUX in front of each of them with the correct operations (with A1A0 as selectors)

Ex. Design a multifunction register as specified in the table using T flip flops.

c1	**c0**	**Function**
0	0	Store registers contents
0	1	Left Shift (serial input connected to input I)
1	0	Count Down

Store: The store is easy, since T ff, we just feed in 0.

Left Shift We are not sure about this one. It would be easy with a D ff, but we are using a T. Whenever we are unsure, we create a table to visualize it.



For the T2 and T1, it is self explanatory. For T0, we need the XOR because again we are using T ffs not D ffs. It makes sense, just trust me.

Now for the T2 and T1, we need an equation so use 2 k maps to get:

$$T_1 = Q_1 \oplus Q_2$$
$$T_2 = Q_2 \oplus Q_1$$
$$T_0 = I \oplus Q_0$$

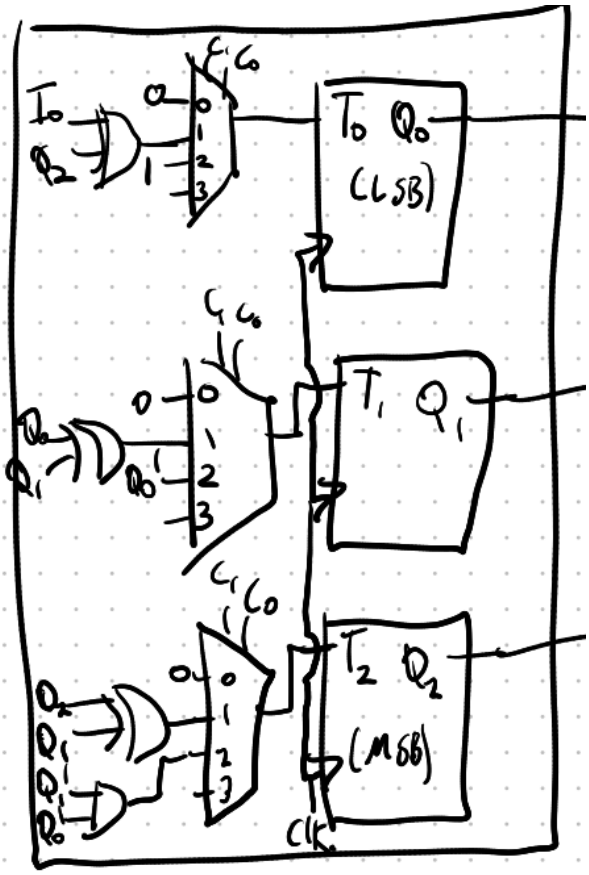
YAY!! Now we just need 1 more ops.

Count Down: This is the same process as the shift left. We just need to create the state table, and then find the values for T2T1T0 using K maps:

$$T_0 = 1$$
$$T_1 = Q_0'$$
$$T_2 = Q_1'Q_0'$$

Now we are done. We know how to do each of the 3 operations, we just need to *choose* between them. Since we see **choose**, we know MUX.

We can ignore the last input of our 4x1 muxs since we do not need it.



Most cases are not this bad since we use D flip flops.

For example, for the left shift using a D, we just put D2 as D1, and D1 as D0, and then D0 as I. SIMPLE!!

2.2 Memory

If we have k address lines in a memory chip and n data output lines, it means we can store 2^k n bit words in that memory chip.

3 Chapter 3

3.1 2s Compliment in Binary

To get the 2s compliment of a number in binary, we flip EACH bit (so 1 becomes 0, 0 becomes 1) and then we add 1 to that number.

Note that when we are using signed numbers where 0 is $+$ and 1 is -, then the 2s compliment of a positive number is itself.

3.2 Overflow

With signed numbers, overflow occurs when the carry into the sign bit and carry out of the sign bit are different. So:

$$OVERFLOW = C_{sign-in} \oplus C_{sign-out}$$

Ex.

Carry	1 0 0 0	
-3	1101	
+ -6	+1010	
-9	1 0111	= +7?

Here overflow occurs since the bit into carry 0 is not the same as the bit out of carry 1.

3.3 IEEE Floating Point

This is a standerdized way of representing floating bit numbers.

We have 3 parts to the 32 bit number to represent a binary number n.

1. Sign (1 bit)
2. Exponent (8 bits) - 127 + The exponent the fractional part needs to be raised to to get n
3. Mantissa (23 bits) - The fractional part of the normalized number n

Ex. Express $(36.5625)_{10}$ as a 32 bit FP number using the IEEE standard.

The value in binary is 100100.1001 and normalized is 1.001001001×2^5.

1. Sign: 0
2. Exponent: 5+127 = 131 = 10000100
3. Mantissa: 00100100100000000000000

So the 32 bit number is: 0 10000100 00100100100000000000000 (without the spaces ofc)

4 Chapter 4

An operation executed in *one* clock cycle is called a **micro operation**.

4.1 Arithmetic Operations

These are things such as adding, subtracting, and really anything that uses an adder block.

4.2 Logic Operations

These are things that are not arithmetic but are still done such as AND, OR, SHIFTS, etc.

A circular shift will shift everything left or right and the spare bit will be taken from the other side.

A logical shift will shift everything left or right and the spare bit will be always a 0.

An arithmetic shift will shift everything left or right and the spare bit if right will be the same as the original MSB, and if left will be 0.

Right Arithmetic shift is division, and Left Arithmetic shift is multiplication.