Department of Electrical Engineering, IIT Bombay

EE206 Digital Circuits: Tutorial Sheet IV

Sequential Logic

1. **Clearing Concepts** *Clearly* explain the significance of an *asynchronous* clear input ($\overline{CLEAR}$) to a positive edge-triggered D-Flip Flop shown in Figure 1(a).

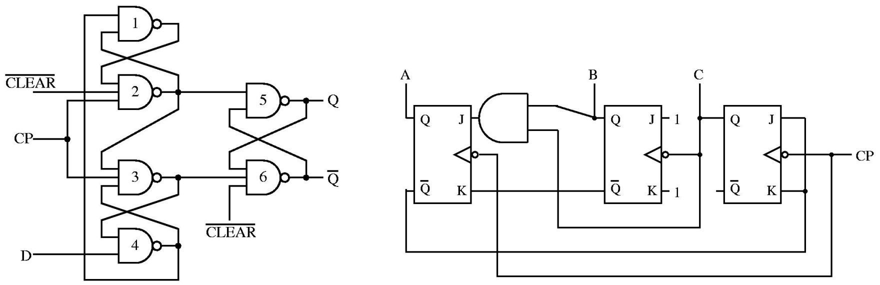


Figure 1: (a) Positive Edge-Triggered D-Flip Flop with Asynchronous Clear, and (b) An example of a ripple counter

2. **Positively Edgy** *Based on Mano, Problem 7-1* Draw the basic unit of a n-bit register consisting of positive edge-triggered D-Flip Flops. The functionalities in the registers are to be an asynchronous $\overline{CLEAR}$ and a $LOAD$ input - both common to all n Flip Flops. Now, modify this structure so that the system is a negative edge triggered system. You can use any SSI hardware for the task.

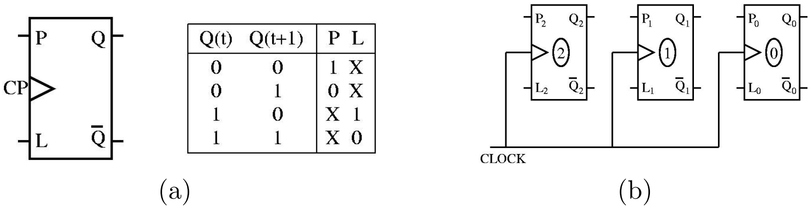
3. *Drill Problem, Mano, Problem 7-4* Design the sequential circuit whose state table is given below using a 2-bit register (constructed using D-Flip Flops), and combinational gates.

Present State		Input	Next State	
A	B	x	A	B
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	0	1

4. **Out of sync?** *Mano, Problem 7-18* The ripple counter shown in the Figure 1(b) uses Flip Flops that trigger on the negative edge transition of the CP input. Determine the count sequence of the counter. Is the counter self-starting? (A counter is said to be **self-starting** if it can start from any state, but eventually reaches the normal count sequence.)

5. **Synchronous Sequential Circuit Design**

   (a) **PQLiar Question** Consider a hypothetical "PL"-Flip Flop, shown in Figure 2(a), along with its *Excitation Table*. Derive the corresponding *Characteristic Table*. Note that this hypothetical PL-Flip Flop has exactly 3 inputs P, L and CP; and exactly 2 outputs Q and $\overline{Q}$.



Q(t)	Q(t+1)	P L
0	0	1 X
0	1	0 X
1	0	X 1
1	1	X 0

Figure 2: (a) PL-Flip Flop, and (b) Figure for the Counter

   (b) **A "Counter Example"** It is required to design a binary mod-5 Synchronous up-counter using PL-Flip Flops, as shown in Figure 2(b). $Q_2Q_1Q_0$ changes as $000 \rightarrow 001 \rightarrow 010 \ldots$ (Q_2 is the MSB, Q_0 is the LSB), such that the machine always functions as an up-counter, and the only way a count can go down, is to 000.

      i. Write down the Excitation Table for the counter.

      ii. Obtain simplified SOP expressions for the inputs P_2, L_2, P_1, L_1, P_0, L_0 in terms of Q_2, Q_1, Q_0, and their complements.

      iii. Implement the combinational logic part of the Counter using a *Minimum* number of 2-input NAND gates *only.*

6. **Mealy and Moore Machines**

   (a) Consider the construction to convert a Mealy Machine M into its equivalent Moore Machine M - this is just a construction, without a proof. Given that M enters arbitrarily numbered states q_0, q_1, … q_n on inputs a_1, a_2, … a_n, and emits outputs b_1, b_2, … b_n, then M' should enter states q'_0, q'_1, … q'_n on the same inputs, and emit the same outputs. Specify q'_0, q'_1, … q'_n. Next, prove the above statement by induction on the length of the input string.

   (b) Convert the Mealy Machine in Figure 3(a) to its equivalent Moore model.

7. **DFAs** Explain your answer in each case. Consider $\Sigma = \{0, 1\}$.

   (a) Construct a DFA to accept strings starting with 1 and representing numbers divisible by 3. (*e.g.* , string 11 would be accepted, whereas strings 0, 00011 and 10 would not: the MSB of these strings is the leftmost bit and the LSB is the rightmost one. The above strings represent the decimal numbers 3, 0, 3 and 2, respectively.

   (b) *Hopcroft & Ullman, Excercise 2-5a* Construct a DFA to accept the set of all strings ending in 00.

   (c) *Hopcroft & Ullman, Excercise 2-5c* Construct a DFA to accept the set of all strings such that every block of 3 consecutive symbols contains at least two 0s.

8. **Being stingy: State Minimization** Consider the DFA shown in Figure 3(b). Write down the 5-tuple Finite State Machine definition for this DFA. Now, perform the k-Equivalence State Minimization Algorithm on this DFA, and show the steps involved.

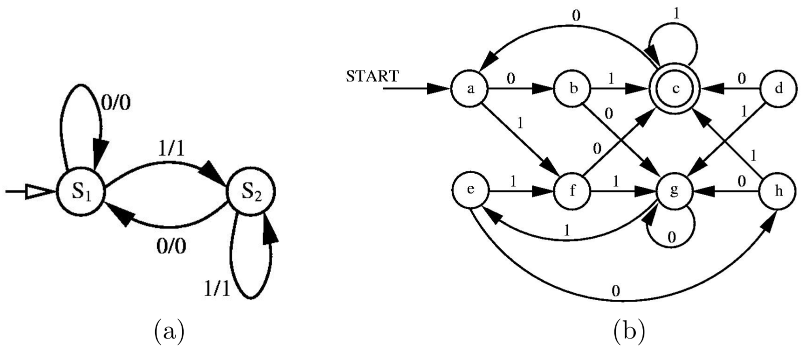


Figure 3: (a) Mealy Machine for conversion to Moore, and (b) DFA for State Minimization

9. **AHPL Question 1** *Hill & Peterson: Digital Systems, Problem 4-11:* Construct a detailed logic block diagram of the hardware realization of the control and data units specified by the following AHPL description, where **a** and **b** are flip flops, **x** is an input, and **z** is a single output line.

```
1 a ← x ∨ b
2 b ← x
  → (a, ā) / (1, 3)
3 z = 1; b ← x ⊕ b
  → (1)
```

Use the convention of one Flip Flop per control step, show the hardware implementation of *both the Data Unit, as well as the Control Unit.* For reference, a sample AHPL program line embodies all concepts required for the above implementation:

```
17 A ← B;
  → (l₁, l₂) / (3, 19)
```

Here, the first part is the data transfer part, which involves a transfer from register **B** to register **A** (Had **B** been a bus, the ← would have been replaced by a =). The second part of the statement is the control portion, which involves a conditional branch to statements numbered 3 and 19, based on the logical expressions l_1 (*e.g.* , $\bar{\mathbf{c}} \vee \mathbf{d}$) and l_2 being 1, respectively. (Unconditional branch to 4 represented as: → 4). An AHPL statement can have a blank control or branch part also.

10. **AHPL Question 2** *Hill & Peterson: Digital Systems, Problem 4-19:* A digital system has an 8-bit vector **X** of input lines together with another input line **ready**. The system will have two output lines **z** and **ask**. Following each one period level placed on line **ask**, a new vector will be available on lines **X**. the availability of such a vector will be signaled by a one-period level on the line **ready**. Each time an input vector is exactly the same as either of the two previous input vectors, the output **z** should be 1 for four clock periods prior to another signal on link **ask**. The line **z** should be 0 at all other times. Write an AHPL sequence describing this digital system. Define the data registers as required.