EMBEDIX: Embedded Computer systems

The Mærsk Mc-Kinney Møller Institute
University of Southern Denmark

Robotronix: Electronics for robotics Fall 2016

Latches, flip-flops and meta-stability

Abstract

The flip-flop is the defining component for almost all digital systems involving some aspect of “memory” e.g.:

- SRAM over
- state machines
- processors

all use flip-flops to remember the state of binary signals from one clock cycle to the next.

The flip flop itself is a bistable circuit, with two stable states – usually called “0” and “1”. This bistable circuit is actually build from logic gates, using feedback to achieve the ability to “remember” the state.

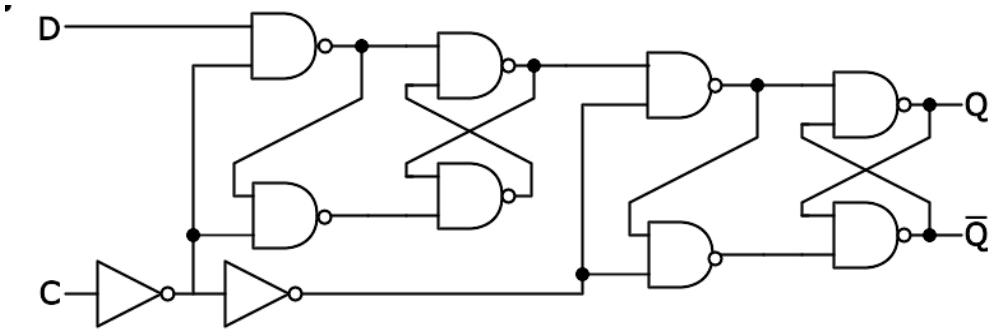


figure 1: D flip flop

During this lab assignment, we will get some basic experience with flip-flops, and also explore some of the lesser known properties that arise from the inner workings of the flip flop.

NOTE!
Metastability CAN ONLY arise when timing requirements (setup & hold time) of flip-flops are violated. Thus it NEVER arise in well designed syncronous systems. The ONLY time we should consider potential metastability is when unsyncronized (random) signals enter a syncronous design. For instance an external input to a state machine.
SO: Metastability is very rare in practice … but fun to play with to get better at building and investigating circuits.

Learning goals:

- More experience with building electronic test circuits
- Understanding of simple un-clocked feed-back logic
  - State transitions in NAND based R-S latch
- Understanding of how to use D Flip-flops
- Understanding of metastability

Goals of the lab assignment

- Implement and document an SR-latch using 74HC00 and 74HC132
  - Document the state transitions for both types
    - Use both y(t) and y(x) measurements on the scope
    - Explain the different behaviour of the two types
    - Provoke and document metastability for both types
  - Document the minimum time gap between S and R in order to avoid metastability

IMPORTANT!! The (grey) goals and reading are NOT mandatory, but has been left in from a previous (more ambitious) version of the exercise, as possible inspiration for you … if you should get bored :-)

- Use a 74HC74 D flip-flop to divide the clock frequency from your oscillator with 2 and 4.
- Try to provoke metastability in a 74HC74 (dual D flip-flop)
  - Using the D and CLK inputs
  - Using the PRE and CLR inputs

Mandatory reading:

- 74HC00 datasheet
- 74HC132 datasheet
- 74HC74 datasheet
- wikipedia on “latch” (electronics)
- wikipedia on “flip-flop” (electronics)
- Texas Instruments: Metastable Response in 5-V Logic Circuits
- Dr. Howard Johnson: Inducing Metastability HIGH-SPEED DIGITAL DESIGN - online newsletter - Vol. 4 Issue 4

Recommended reading (some useful ideas)

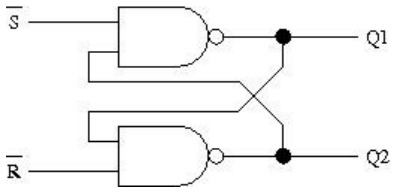
- Altera: metastability in Altera Devices
- Scientific America: Computers without clocks

Additional reading (for the fun of it)

- 3 different papers you mind find interesting

SR latch

It's easy to build the simple bistable memory element known as a 'latch' or Set-Reset flip-flop, using two NAND gates:



(Note that the D flip flop shown in figure 1, has two of these latches inside it.)

Under the assumption that the signals Q1 and Q2 are digital, with only the states {0,1} possible, the circuit can be said to have four states, which we will refer to as the ordered pair (Q1,Q2), so the names of the states become: {00,01,10,11}

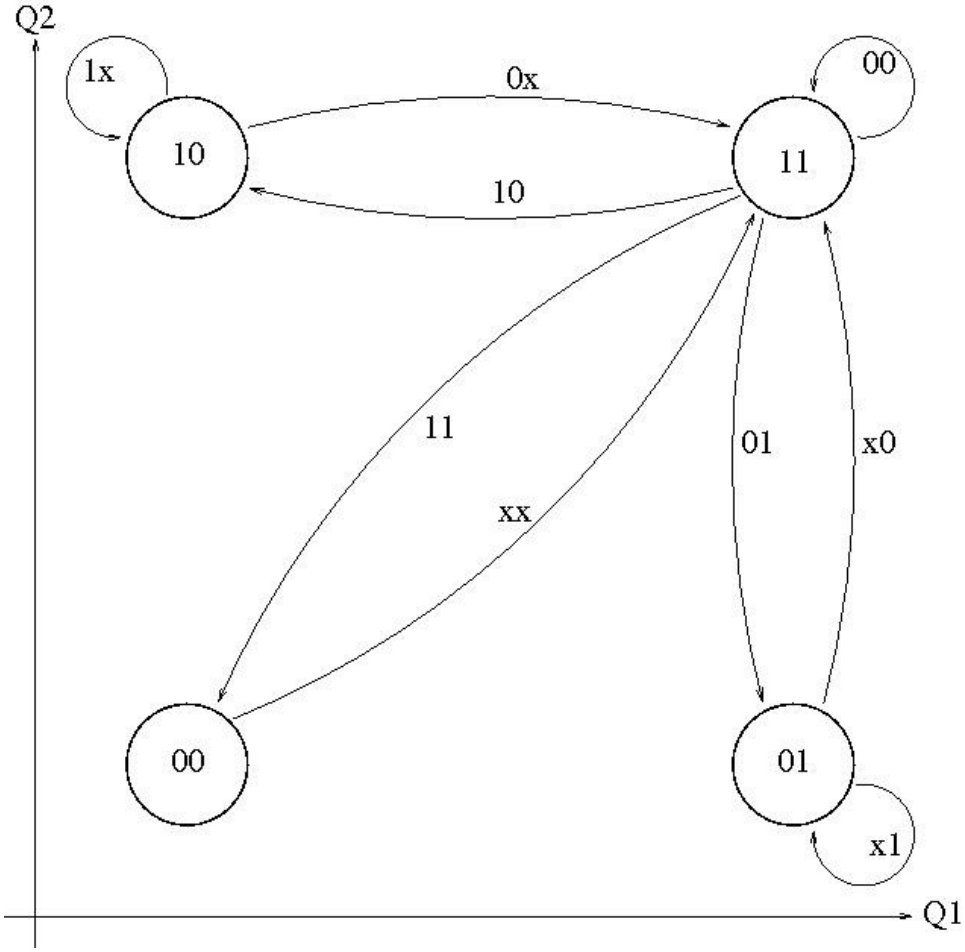
If the inputs are also assumed to be digital signals, we see that there are four possible inputs to the circuit, which we will refer to as the ordered pair (/R,/S), so the four possible inputs become {00,01,10,11}

Making an exhaustive table, that tell the response of the two NAND gates – and hence the “next” state – is then straightforward, and shown below.

State		**Inputs**		**Nxt state**	
Q1	Q2	/R	/S	Q1'	Q2'
0	0	0	0	1	1
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	1	1
0	1	0	1	0	1
0	1	1	0	1	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	1	1
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	0	0

It is somewhat easier to view this information as a directed graph, with four nodes, representing the four states, and edges representing the transition from current- to next-state.

The transitions are labeled (/R,/S), with the notation “x1” used if the same transition will happen whether /R = 1 or /R = 0.



As /R and /S are 'active low' signals, their normal (passive) state must be assumed to be “11”. And the normal states for the latch to be in will be either “10” or “01”

These two states are clearly stable when the input is “11”. If the circuit is in state “10” and the input becomes “01”, then the circuit will leave the state, and stabilize in state “01”. If the input returns to “11”, the circuit will remain in that state.

If the input should change to “10”, the opposite transition will happen, and the circuit returns to “10” again.

If both /R and /S are active, that is “00”, the circuit will be forced to stay in the “11” state, until the input changes. If the input changes to “11”, then it seems the circuit will go to the “00” state”, except the “00” state is unstable, and the circuit will immediately abandon state “00” no matter what the input is, and return to “11”, where the same will happen again, as long as the input is “11”.

This predicted behaviour sounds quite interesting, and worth some investigation! It will be particularly interesting to see how the assumption of Q1 and Q2 as digital (two level) signals will compare to the real world of continous signals.

LAB work

1. Build two similar SR latches using a 74HC00 for the first, and a 74HC132 for the second. Document and compare their behaviour to all combinations of states and inputs.

   Tip: If you connect the two channels of the scope to Q1 and Q2 respectively, and set the scope in X/Y mode, the scope will plot Q1 along one axis and Q2 along the other axis, allowing you to trace the relationship between Q1 and Q2 during state shifts. Each corner of the screen will thus correspond to one of the four states.

   a) How does the SR latches actually switch between the stable “01” and “10” states, when /R and /S are activated separately

      TIP: You can activate them (almost) separately by connecting a square wave to /R and using an inverter to make an inverted counterpart of the square wave to drive /S. In this way, the input will always shift between “10” and “01”

   b) What actually happens if /S and /R are activated simultaneously

   c) Can we achieve metastable behaviour by releasing /R and /S simultaneously? Document the metastable behaviour using both X/Y plots and ordinary plots of Q1 and Q2 as a function of time.

      TIP: As the gates within a package are not completely identical, and as your wiring – and thus parasitic components – are probably not completely symmetrical, it may be necessary to introduce a slight skew between /R and /S to achieve metastability. Use your variable skew generator from last time to drive /R and /S. You probably get the best results if you drive it with a slow clock frequency (less than 1 Mhz) so you get time to observe the reaction before a new clock pulse comes along.

   d) For the above points: Explain the behaviour of the circuits, and the difference between the 74HC00 and 74HC132 version of the circuit.

2. Experiment with the 74HC74 Dual D flipflop

   a) Build a 1 bit counter by connecting the /Q output of a 74HC74 D flip-flop to its own D input. In this way, the flip-flop will change state each time there is a rising edge on the CLK input. The pre and clr inputs must be held inactive.

      Document the behaviour of the counter, including a plot of the input and output signals, as well as a X/Y plot of the state transitions between the Q and /Q outputs. Document any time delay from input edge to output edge.

Comment on any interesting observations or parallels to observations of the SR latch.

b) Expand it to a 2 bit counter, by adding an extra flip-flop, and demonstrate that the input frequency can now be divided by 4.

   Document any delays between input edges and the two output edges (the divide by 2 and divide by 4 outputs), and also any delays between the two different output edges.

   Document anything else you find interesting.

3. Try to get the 74HC74 flip flop metastable: (VERY VERY DIFFICULT!!!)

   a) By using your adjustable skew to drive the D and clk inputs almost simultaneously. Adjust the skew to maximize the probability of a metastable response

      TIP: It is quite difficult to do so a great deal of patience is probably required. In order to catch the metastable behaviour, you will probably need a storage scope, using one-shot trigger, and triggering manually until you get lucky.

   b) By using your adjustable skew to drive the pre and clr inputs almost simultaneously.

   c) Document metastability (if you can achieve it) with X/Y plots of the Q and /Q outputs, as well as by normal plots of Q(t) and /Q(t), for bothe of the above experiments. Compare your findings to your experience with the SR latch.

   d) Compare the skew at which you obtained metastability with the 74HC74 data sheets specifications of set-up and hold time and other relevant data.