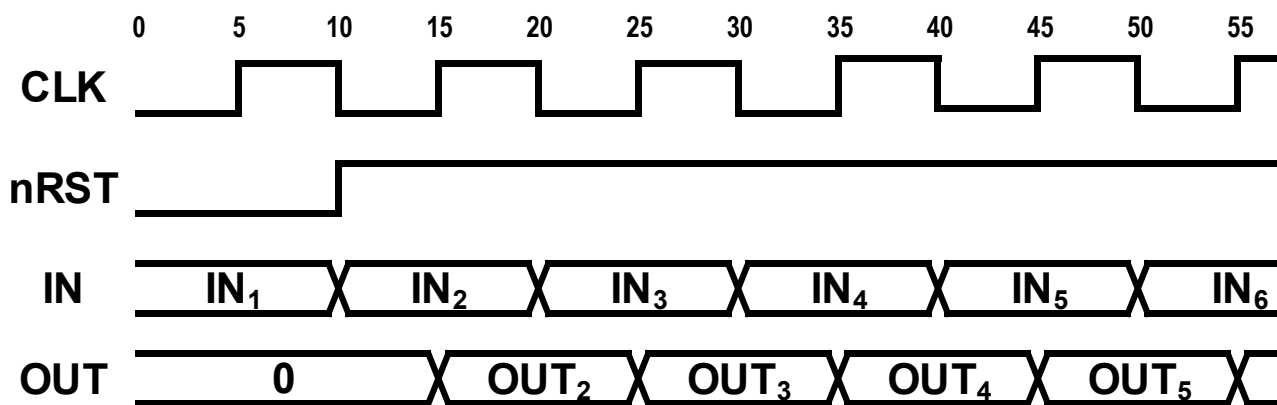


# In Class Assignment #5 (1) (10/16)

- **Design a 4-bit comparator and a BCD to decimal decoder using behavioral modeling**
  - Make each output to be captured by a positive edge D flipflop with negative edge triggered asynchronous reset (CLK, nRST)
  - Use at least two always statements for comb/seq circuit each
  - Set the clock period of 10
  - Use the named mapping to instantiate each design
  - Input stimulus and the expected output timing diagram



- Submit zip files that contain the source code and screen capture to LMS

# In Class Assignment #5 (2) (10/16)

## ■ 4-bit comparator (DD05-1\_StudentID.v)

- Module: MyComparatorBeh, Stimulus: Top
- Input: A, B (4-bits), CLK, nRST, Output: A\_lt\_B, A\_gt\_B, A\_eq\_B (1-bit)
- Apply the input pairs (A, B) = (4'h0, 4'h1), (4'hf, 4'h3), (4'ha, 4'hb), (4'h0, 4'h0), (4'h7, 4'hd), (4'h9, 4'h9)

```

0 A = 0, B = 1, A_lt_B = 0, A_gt_B = 0, A_eq_B = 0
10 A = f, B = 3, A_lt_B = 0, A_gt_B = 0, A_eq_B = 0
15 A = f, B = 3, A_lt_B = 0, A_gt_B = 1, A_eq_B = 0
20 A = a, B = b, A_lt_B = 0, A_gt_B = 1, A_eq_B = 0
25 A = a, B = b, A_lt_B = 1, A_gt_B = 0, A_eq_B = 0
30 A = 0, B = 0, A_lt_B = 1, A_gt_B = 0, A_eq_B = 0
35 A = 0, B = 0, A_lt_B = 0, A_gt_B = 0, A_eq_B = 1
40 A = 7, B = d, A_lt_B = 0, A_gt_B = 0, A_eq_B = 1
45 A = 7, B = d, A_lt_B = 1, A_gt_B = 0, A_eq_B = 0
50 A = 9, B = 9, A_lt_B = 1, A_gt_B = 0, A_eq_B = 0
55 A = 9, B = 9, A_lt_B = 0, A_gt_B = 0, A_eq_B = 1

```

```

0 BCDIn = 0 --> DECOut = 0000000000
10 BCDIn = 1 --> DECOut = 0000000000
15 BCDIn = 1 --> DECOut = 0000000010
20 BCDIn = 2 --> DECOut = 0000000010
25 BCDIn = 2 --> DECOut = 0000000100
30 BCDIn = 3 --> DECOut = 0000000100
35 BCDIn = 3 --> DECOut = 0000001000
40 BCDIn = 4 --> DECOut = 0000001000
45 BCDIn = 4 --> DECOut = 0000010000
50 BCDIn = 5 --> DECOut = 0000010000
55 BCDIn = 5 --> DECOut = 0000100000
60 BCDIn = 6 --> DECOut = 0000100000
65 BCDIn = 6 --> DECOut = 0001000000
70 BCDIn = 7 --> DECOut = 0001000000
75 BCDIn = 7 --> DECOut = 0010000000
80 BCDIn = 8 --> DECOut = 0010000000
85 BCDIn = 8 --> DECOut = 0100000000
90 BCDIn = 9 --> DECOut = 0100000000
95 BCDIn = 9 --> DECOut = 1000000000
100 BCDIn = 10 --> DECOut = 1000000000
105 BCDIn = 10 --> DECOut = 0000000000
110 BCDIn = 11 --> DECOut = 0000000000
120 BCDIn = 12 --> DECOut = 0000000000
130 BCDIn = 13 --> DECOut = 0000000000
140 BCDIn = 14 --> DECOut = 0000000000
150 BCDIn = 15 --> DECOut = 0000000000
160 BCDIn = 0 --> DECOut = 0000000000
165 BCDIn = 0 --> DECOut = 0000000001

```

## ■ BCD to decimal decoder (DD05-2\_StudentID.v)

- Module: BCDtoDecimalBeh, Stimulus: Top
- Input: BCDIn (4-bits), CLK, nRST
- Output: DECOut (10-bits)
- Apply the input from 0 to 15 and 0 again