Homework # 4

Due Thur., Nov. 14, 2014, at 5pm. No late homeworks will be accepted except for verifiable illness or similar situations.

1. **From Latches to Flip Flops**

   (a) Build a **gated** D **latch** from an (ordinary) R-S latch. It has two inputs, D and G, and an output Q. Recall that for a gated D latch:

   - When $G = 1$ and $D = 1$, Q is set to 1.
   - When $G = 1$ and $D = 0$, Q is set to 0.
   - When $G = 0$, Q holds its last value.

   [*5 points*]

   (b) Build a **rising edge-triggered** D flip-flop from a pair of gated D latches. [*5 points*]

   (c) Build a **falling edge-triggered** D flip-flop from a pair of gated D latches. [*5 points*]

   (d) Build a T **flip-flop** from a D flip-flop. [*5 points*]

   (e) Build a T **flip-flop** with *Clear* input. [*5 points*]

2. **Counter/Register Combination**

   Design a circuit as follows. It has a four-bit counter with a *Clear* input. It also has a four-bit register that grabs the outputs of the counter when a *Grab* input is high (and holds these values when the Grab input returns to low). So the inputs are:

   - *Clock*.
   - *Clear* (When high, the counter should set the four bits of the counter to zero on the next rising edge of the clock.)
   - *Grab* (When high, the register should grab and hold the four bits of the counter on the next rising edge of the clock.)

   The outputs are:

- Four bits from the register: D_0, D_1, D_2 and D_3.

Design your circuit with logic gates, D flip-flops and and T flip-flops. (You may assume that that the D and T flip-flops have *Hold* and *Clear* inputs, respectively.)

3. **Mealy Machine**

   A sequential circuit has one input X and two outputs Z_1 and Z_2. An output $Z_1 = 1$ occurs every time the input sequence 100 is complete provided that the sequence 011 has never occurred. An output $Z_2 = 1$ occurs every time the input 011 is complete. Note that once a $Z_2 = 1$ output has occurred, $Z_1 = 1$ can never occur but *not* vice-versa. Also, the bits of X are interpreted as a *sliding window*.

   - Draw a Mealy state diagram. (Mealy means that the outputs depend on both the current state and the input.)
   - Assign binary labels to the states and write the truth tables for the next state and the output.

4. **Moore Machine**

   Design a sequential circuit with a single input X and two outputs Y and Z with the following behavior:

   - In the "*Idle*" state, it should output $Y = Z = 1$.
   - So long as it receives zeros on X, it should remain in the *Idle* state.
   - When it receives a one on X, it should output $Y = 0$, $Z = 1$. It should continue to output these values so long as it receives zeros on X.
   - When it receives a second one on X, it should output $Y = Z = 1$. It should continue to output these values so long as it receives zeros on X.
   - When it receives a third one on X, it should output $Y = 1$, $Z = 0$. It should continue to output these values so long as it receives zeros on X.
   - When it receives a fourth one on X, it should go back to the *Idle* state.

   Design your circuit as a **Moore** machine (so the outputs Y and Z depend on the current state, but not on the input.)

   (a) Draw a state diagram.

   (b) Provide truth tables for the bits of the next state.

   (c) Provide truth tables for the outputs.

   (d) Implement the functions for the bits of the next state and the outputs as two-level AND-OR circuits, using Karnaugh maps.