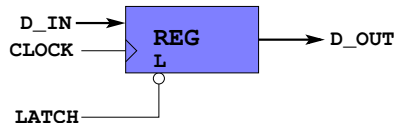


## EE 231L Lab 3

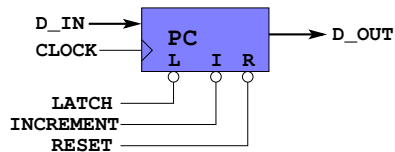
## Prelab Part 3

For the final computer you will need two types of registers. One type of register is a simple synchronous latch:



When LATCH is high, the output data D\_OUT will not change. When LATCH is low, the input data D\_IN should be latched into the register on the rising edge of CLOCK.

The second type of register is called a program counter (PC). This keeps track of which instruction in memory to execute. Usually programs are executed sequentially, so after executing the instruction at address, say, 0x0123, the program will then execute the instruction at address 0x0124. In this case PC needs to increment after each instruction is executed. Sometimes the program needs to execute code in a different area of memory – flow control statements such as `for` and `while` do this. In this case, the PC needs to be loaded with a new address. In order for the program to start, you will need to reset the program counter to zero to start execution at the first instruction of the program. Here is what the PC looks like:



Normally, INCREMENT, LOAD and RESET will be high. When INCREMENT is low, the PC should increment D\_OUT to  $D\_OUT + 1$  on the rising edge of CLOCK. When LATCH is low, the input data D\_IN should be latched into the register on the rising edge of CLOCK. The system which controls PC will ensure that LATCH and INCREMENT are never low at the same time. (In your program, you should have PC do something sensible, like latch D\_IN, if both happen to be low simultaneously.) When RESET is low, PC should immediately reset to 0x00; it shouldn't wait for a clock edge. This is normally called a synchronous counter with synchronous load and asynchronous reset.

1. Design an eight-bit synchronous latch in AHDL.
2. Design an eight bit program counter as described above in AHDL.
3. Write a program which uses the above two designs as functions to test that they work.