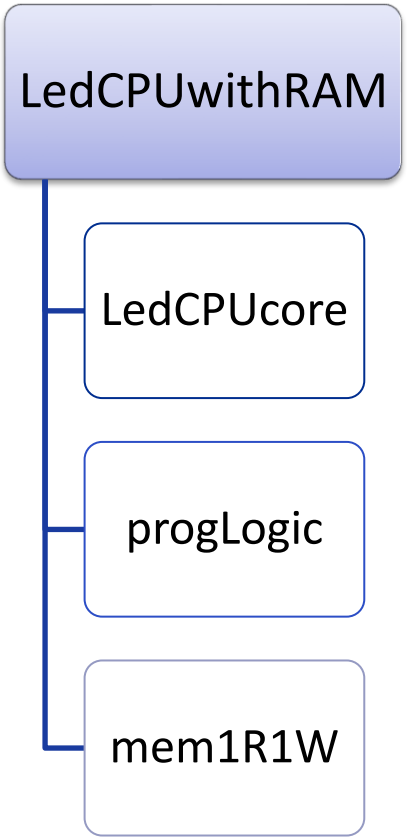
Led CPU With RAM

Design Description:

- It reads instructions from RAM instead of ROM, and executes these instructions.
- The difference between ROM and RAM CPU is that RAM CPU is programmable.
- It should execute 2 different commands like ROM CPU. These are jump and delay.

Hierarchy of LedCPUwithRAM



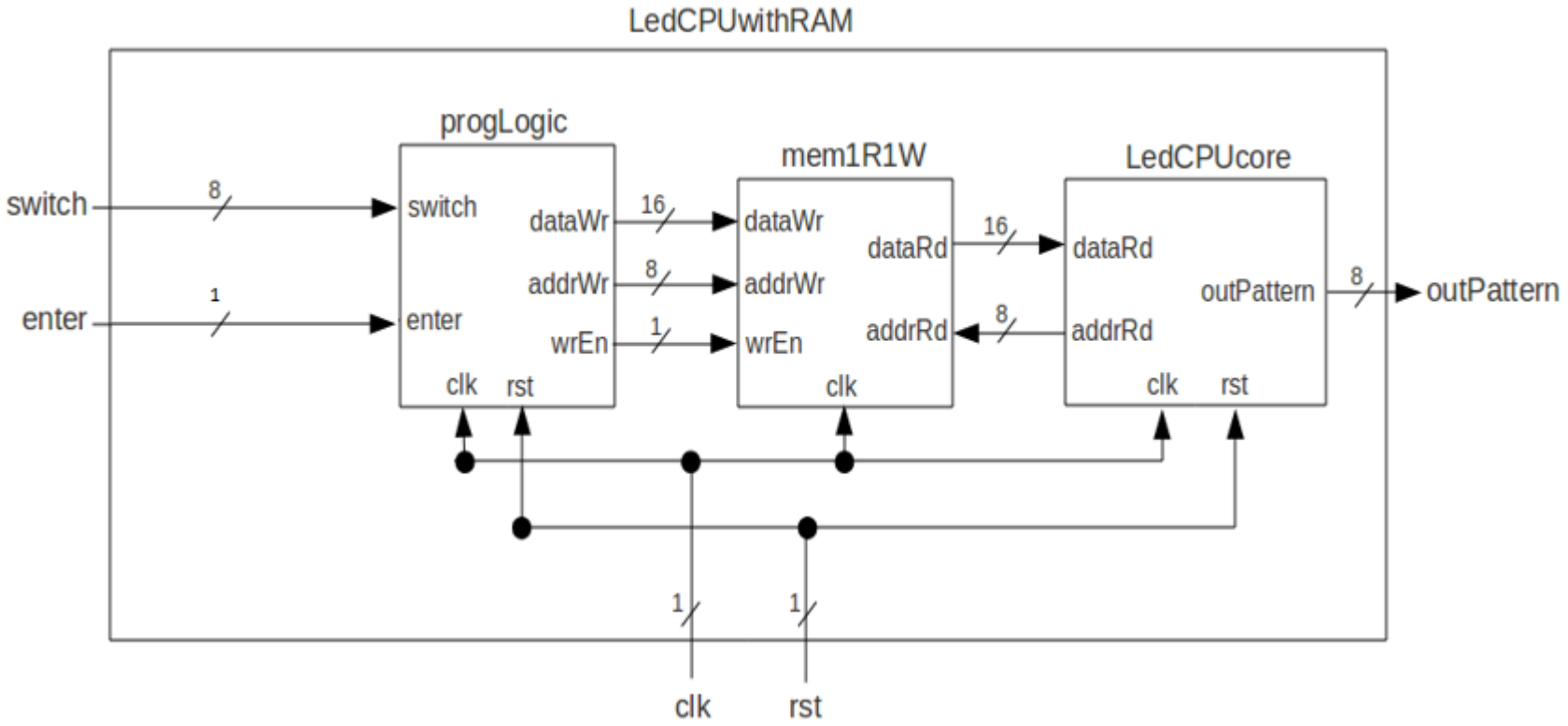
Top module of the design and it only contains instantiation of the other modules. You will use existing code for this module.

This module reads instruction from either RAM or ROM and drives LEDs . You will use existing code for this module.

This module is to program LedCPUwithRAM while filling RAM using switches and a push button. **You will write source code for this module.**

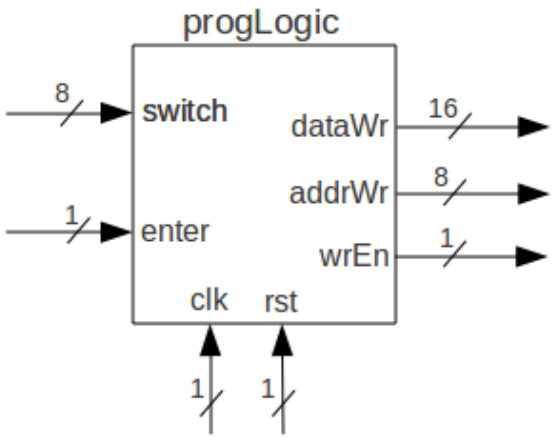
This module is RAM module.
You will use existing code for this module.

Block Diagram of LedCPUwithRAM



Design I/O:

clk : 1 bit input for clock
rst : 1 bit input for reset
switch : 8 bits input comes from switches as data in
enter : 1 bit input comes from a push button when data is ready on switches
dataWr : 16 bits data output to RAM
addrWr : 8 bits address output to RAM
wrEn : 1 bits output as write enable to RAM



Design Behavior:

- The aim of this module is to fill RAM. RAM entries have 16 bits of length, and we will use 8 bits switches to fill it. So, we should first take 8 bits and then second 8 bits from switches.
- We will use an enter signal, which comes from a push button, as data ready signal.
- When the enter signal goes from 0 to 1 for first time, the data on switches will be our first 8 bits part of the 16 bits data. When for the second time the enter signal goes from 0 to 1, the data on switches will be our second 8 bits part of the 16 bits data.
- When 16 bits of data is ready, we should send this data to RAM while setting address location. Initial address location is 0 and it should increment 1 after 16 bits data was sent to RAM.
- Meanwhile wrEn signal should be 1 when we want to write data to RAM. Otherwise it should be 0.