



- Inputs
  - wire clock;      // 12.5 MHz clock; **MUST** be same as imem clock
  - wire enable;      // set to 1 when you want to load a new program
  - wire rx;      // UART wire PIN C4 (Nexys 4) on the board in your constraints file
  - wire cts;      // UART wire PIN E5 (Nexys 4) on the board in your constraints file
- Outputs
  - wire tx;      // UART wire; PIN D4 (Nexys 4) on the board in your constraints file
  - wire rtr;      // UART wire; PIN D3 (Nexys 4) on the board in your constraints file
  - logic wr\_imem;      // connect to write enable port on imem\*
  - logic [31:0] addr\_imem;      // connect to write address port on imem\*
  - logic [31:0] data\_imem;      // connect to write data port on imem\*
  - logic [2:0] state;      // do what you want with it, it conveys the internal state of the programloader

\* You must convert your imem to a read/write memory and not just the read only it currently is. I leave that exercise to you.

\* When you have enable turned on, you **MUST** have your MIPS in reset so its program counter stays at 0 and it isn't executing instructions

\* rx, cts, tx, rtr **MUST** be as input/output wires on your top level module going to the board itself

- Once you have the module put into your hardware and the appropriate connections made, burn the design to the board
- Install FTDI CDM21224 driver on your computer (Windows). This allows your USB to be operated as a serial port. Zip file is in the Google Drive folder. The link is also here <http://www.ftdichip.com/Drivers/VCP.htm>
- When you want to load a program ensure the enable wire is set to 1
  - I used one of the switches to send this enable signal so when I want to reprogram I just flip the switch
- I have written a companion program in C# that you must run on your PC to actually send the new program over the USB wire to your board. Written and tested in Windows but if there is a demand I can get it working on Linux. No Apple will be available as I don't have a MAC to compile with (not even sure it can handle C#) and our Vivado doesn't run on Apple anyway.

- Run the companion program (progloader.exe) on your PC. The program is also located on the Google Drive. Command line arguments are the COM port number and the filename. You will need to get what COM port was assigned to your USB connection (usually COM3 or COM4). The filename is the already compiled instructions from the MARS assembler.
- Then follow the directions the programs gives you. Mostly just press a key to continue or ensure your hardware is set to program load.
- All instructions will be sent serially over the USB to your hardware and if you have done it correctly, it will load all the instructions in imem and start executing them once you set the enable back to 0.
- Not going to hand hold you through it but if you have questions or need a little help then let me know via email or before/after class. I usually get to the classroom about 1030 and can stay after if needed ([djsteffey@cs.unc.edu](mailto:djsteffey@cs.unc.edu))