

## Checklist

### **Inside CPU:**

1. Microsequencer
  - This module controls the instruction counter, which determines the next instruction to be read from memory. It handles jumps and (un)conditional branches (absolute and relative, respectively), maskable interrupts, and subroutine calls and returns with an onboard instruction counter stack (separating the instruction counter from the normal datapath).
  - There are a number of instructions in the instruction set (see above) which make use of the microsequencer, and they will all be executed in the demonstration (as well as the maintenance operation to reset the IC stack, if it overflows or underflows, if time permits).
2. Register File Manager
  - This module manages the temporary variables in both the word-width and the byte-width register files. Outside the module, each register file appears to have two read ports and one write port, although internally, the capabilities of byte swap and the split-word/join-bytes mechanism are also supported.
  - Nearly every instruction requires the basic functionality of this module, but the special operations it can perform will also be used.
3. ALU Manager
  - This module contains both the word-width and byte-width ALUs, and each ALU performs various operations: add, subtract, logical shift, arithmetic shift, rotation, and the standard bitwise Boolean operations (and, or, xor, nand, nor, xnor). Additionally, the manager module can also perform two special operations in either datapath: bitwise reversal, and bitwise interleaving.
  - The ALU manager's functionality will be demonstrated by performing all of the operations which it supports.
4. Trap
  - This module uses pipelining to handle operations (multiply, divide, and modulus for both words and bytes) which would otherwise unduly increase the minimum clock period of the CPU. Through handshaking signals, trap operations are performed while the microsequencer is disabled, and then the microsequencer is enabled again when the operation has finished.
  - As with the other modules, the trap module will be tested via the operations it performs (although, for demonstration purposes, one of the LEDs may be used to show that the microsequencer has been disabled as well).

## 5. Control Logic

- This module generates the control signals for every other module, both inside and outside the CPU. It also internally supports a mode flag for distinguishing between system mode and user mode, and forbidding system-level operations to be run in user mode (by performing no operation).
- Every operation not only requires the control logic, but is in fact *defined* by the control logic, so this module will be demonstrated by the normal operation of the computer.

### Outside CPU:

## 6. Stack

- This module is a synchronous word-width hardware stack implementation for use by the CPU as a data storage facility.
- The two operations performed by a stack, push and pop, will be demonstrated by execution (as well as the maintenance operation to reset the stack, if it overflows or underflows, if time permits).

## 7. Timer

- This module is a word-width programmable timer, which provides the CPU with the time in milliseconds.
- The set-time and get-time operations will be used to show that a) the timer is programmable, and b) the timer keeps track of the time properly.

## 8. Random

- This module uses a number of ring oscillators to generate a source of random bits, and a von Neumann corrector to produce significantly better random data, and provides a word's worth of random data to be extracted by the CPU.
- Executing the rand operation and showing the resulting data will demonstrate that the random module is generating random data.

## 9. Memory

- This module is the main memory of the computer, with one read port for instruction fetch and one read/write port for data storage and retrieval.
- Every instruction executed will be read from the instruction port, and all load and store operations will use the data port.

## 10. I/O

- This module contains two pre-built modules, ps2 and vga, for controlling a keyboard and a monitor, respectively. It interfaces with the CPU with a word-width datapath and a port-signaling system (to determine with which device the CPU is attempting to communicate). If time permits, an xvga module may be added with a ZBT RAM for video memory to allow graphical display modes, instead of only text mode for the vga.
- The peripheral devices controlled by this module are the user interface for the computer, so normal operation of the computer will show their functionality.