This processor finishes most instructions in two clock cycles with only LDM and STM taking three cycles.

Accumulator: register that accumulates values from alu/shifter operations and is always used as an operand

ALU: Implements ALU instructions: pass, and, or, and, sub, not, inc, dec.

Controler: reads in status signals from datapath and outputs controls based on the instruction input. Uses a counter to implement cycle states.

Count2b: two bit synchronous resetting counter with a stop signal. This is used to keep track of which cycle the current instruction in in.

Datapath: Connects all non-control units together, outputs status signals.

Incer: PC incrementer that outputs PC +1

IRreg: the instruction register used to hold the current instruction. Asynchronous reset with load signal.

Memfile: This file holds the binary equivalent assembly code to the test provided.

Mux4: four input mux for the input to PC

Mux5: Mux input to accumulator last input is redundant but was listed in data path.

Outreg: output register with asynchronous reset, and an output signal that only outputs if high

Pcmux2: two input mux used for the output of pc to the memory address input

Pcreg: PC register with asynchronous clear

RAM: instruction and data memory register file. Updates on falling clock edges when write enable is active.

Regfile: register file for temporary memory storage.

Reln/relp: adds or subtracts the relative jump value from the PC

Shifter: implements all shift operations.

Testbench: calls the top. Gives in clock and reset signals.

Top: connects the datapath and controller. Outputs debugging info for the testbench