3 Stage Pipeline

Group 9:
CS14B023 Rahul Kejriwal
CS13B005 Bharat Sai Botta
CS16S033 Debanjan Ghatak

1. Work Split:

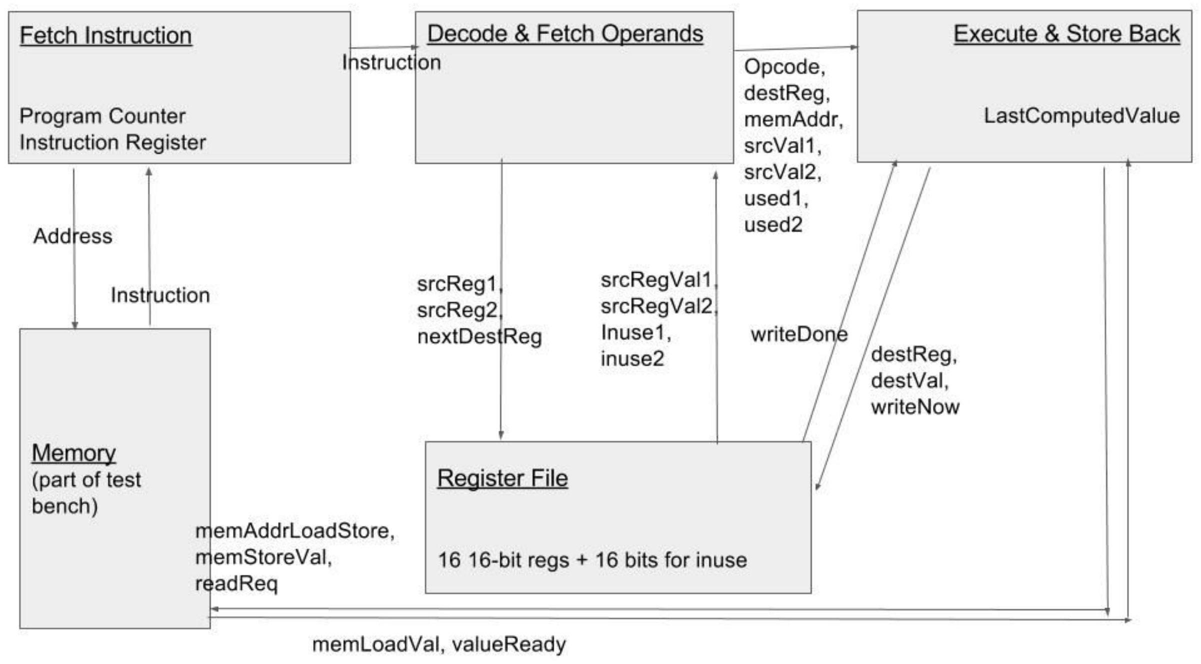
Team Member	**Work Share**
Rahul Kejriwal	1. Designed architecture and module interfaces. 2. Building module for ‘Register File’ unit. 3. Building module for ‘Decode Instruction and Fetch Operands‘ unit. 4. Building corresponding testbenches. 5. Helping Ghatak for ‘Execute and Store Back’ unit.
Bharat Sai Botta	1. Building module for ‘Fetch Instruction’. 2. Building corresponding testbenches.
Debanjan Ghatak	1. Building module for ‘Execute and Store Back’ unit. 2. Building corresponding testbenches.

2. Assumptions:

a. We assume that the ‘Fetch instruction’ unit and the ‘Execute and Store Back’ (for LOAD/STORE instructions) don’t use the same memory address in the same cycle. [As PS says one read/write port per memory address]
b. All units finish their entire work within one clock cycle. [What this means, is that say ‘Decode Instruction and Fetch Operands‘ takes 5ns and ‘Execute and Store Back’ takes 10 ns then the cycle width/duration is greater than 10ns. The cycle duration is large enough for each unit to finish their respective work for their current instruction within that cycle itself.]
c. We are not using any delays (#n) in the verilog code. We do not worry about balancing the time taken by each pipeline stage to improve throughput.
d. Register values can be changed during a cycle (not only on posedges). But register status bits change only on posedges.

3. System Architecture:

a. Here, is the system architecture schematic showing the interfaces between the modules:



Fetch Instruction
Program Counter
Instruction Register
Instruction
Decode & Fetch Operands
Opcode, destReg, memAddr, srcVal1, srcVal2, used1, used2
Execute & Store Back
LastComputedValue
Address
Instruction
srcReg1, srcReg2, nextDestReg
srcRegVal1, srcRegVal2, Inuse1, inuse2
writeDone
destReg, destVal, writeNow
Memory (part of test bench)
memAddrLoadStore, memStoreVal, readReq
Register File
16 16-bit regs + 16 bits for inuse
memLoadVal, valueReady

b. Description of Modules (and variables used in the schematic):

Module	Core Components	Input	Output
Fetch Instruction Unit	1. Program Counter 2. Instruction Register	1. Instruction (to be executed)	1. Instruction Address (to be read) 2. Instruction (transfer to next unit)
Decode & Fetch Operand Unit		1. Instruction (to be decoded) 2. Value of 2 source registers, srcRegVal1 and srcRegVal2 3. Status of the 2 source registers, i.e., whether they are being computed by the Execute Unit currently, inuse1 and inuse2	1. Source register address, srcReg1 and srcReg2 (to be read) 2. Destination register address, nextDestReg (to set inuse bit) 3. Opcode of instruction, opcode 4. Destination register address, destReg for next instruction 5. Source register values, srcVal1 and srcVal2 6. Address of memory location in case of LOAD/Store 7. Inuse bits for source registers, used1 and used2

Register File	1. 16-bit registers x16 2. Inuse bits x16	1. Source register address, srcReg1 and srcReg2 (to be read) 2. Destination register address of next instruction (to set inuse bit) 3. Destination register address of current instruction, destReg and value to be stored there, destVal	1. Value of 2 source registers, srcRegVal1 and srcRegVal2 2. Status of 2 source registers, i.e., whether are being computed by the Execute Unit currently, inuse1 and inuse2
Execute & Store Back Unit	1. LastDestVal (stores the value computed in the last cycle)	1. Opcode of instruction, opcode 2. Destination register address, destReg for next instruction 3. Source register values, srcVal1 and srcVal2 4. Address of memory location in case of LOAD/Store 5. Inuse bits for source registers, used1 and used2 6. Memory read value for LOAD instruction	1. Memory write value for STORE instruction 2. Memory address for LOAD/STORE instruction 3. Destination register address, destReg 4. Destination register value, destVal

c. writeNow, writeDone and readReq, valueReady are synchronization signals between Execute unit and Register File and Memory respectively.
d. Processor Status Word is maintained outside the modules in the top level module Processor. It’s value changes only at posedges.
e. Opcode maps:
   i. 0000 - NOP
   ii. 0001 - HLT
   iii. 0010 - ADD
   iv. 0011 - SUB
   v. 0100 - MUL
   vi. 0101 - SL
   vii. 0110 - SR
   viii. 0111 - AND
   ix. 1000 - OR
   x. 1001 - NOT
   xi. 1010 - XOR
   xii. 1011 - Unused
   xiii. 1100 - Unused
   xiv. 1101 - Unused
   xv. 1110 - Load
   xvi. 1111 - Store