Cairo University
Faculty of Engineering
Computer Department

Computer Architecture

Submitted to: Dr.Ahmed Morgan

Team names:

- Abdallah Magdy
- Amr Ragab
- Yasser Farouk

Design Assumptions:

- The pipeline consists of five stages (fetch, decode, execute, memory access & write back)
- The memory used is Harvard architecture memory, one separate memory for instructions and another one for data to avoid structural hazard between fetch and memory access stages.
- Unknown opcodes make the processor throw an exception and jump to ISR
- The registers values are written in the first half of the clock cycle and read in the second half to avoid structural hazard between decode and write back stages.
- Data hazards are handled by implementing forwarding
- There is a stall after any instructions reads a value from data memory and has data hazard with its next instruction
  ex: LDD R2,3
  DEC R2
- Predict untaken technique is used in B-Format instructions.
- Interrupt handling flushes only the instruction that exist in fetch stage and save its address to begin with after return from interrupt. All the others instructions in the pipeline run as usual.

Design modules:

1- Register file

- Contains the four general purpose registers (R0, R1, R2, R3 (SP))
- It allows writing on two registers – one of them must be R3 - simultaneously.

2- ALSU (Arithmetic logic shift unit)

- It performs the following operations:

ALSU selection input	operation
0000	F = 0
0001	F = B
0010	F = A+B
0011	F = A-B
0100	F = A and B
0101	F = A or B
0110	Rotate lift B with carry
0111	Rotate right B with carry
1000	F= not B
1001	F= negative B
1010	F= B + 1
1011	F= B - 1

3- Data memory & instructions memory

- Each of them are 256 byte size, byte addressable and the same speed as processor (theoretical)
- We can read only – can't write - from instruction memory.

Pipeline stages:

1- Fetch stage:

- Contains the PC and contains the instruction memory
- Reads one byte as an instruction and puts the current PC and IR into IF_ID register

2- Decode stage

- Contains the control unit that:
    - Decodes the coming instruction
    - Generates control signals to all the modules and stages depending on the instruction
    - Calculates the next PC and return it to fetch stage to get the next instruction
    - If a stall happened after load/pop instruction due to data hazard, the instructions that exist in the fetch & decode stage remains in the same place, the execute stage receives a bubble (NOP) from decode stage.
    - Responsible for handling interrupts and exceptions (unknown opcode)

3- Execute stage:

- Contains ALSU and is responsible for changing flags
- Reading the value of IN_PORT in IN instruction
- Contains forwarding logic (get the forwarded values from memory access or write back)
- Calculate branch target address and check if the branch is taken or not
- Checks for stalling need after load instructions

4- Memory access stage

- Contains data memory unit.
- Is responsible for decrementing SP after push instruction

5- Write back stage:

- Write the data required to be written in registers.
- Write the data in OUT_PORT in OUT instruction.

Bonus requirements

- The processor is synthesized successfully with Quartus but it didn't report the clock frequency, so we used Synopsys for synthesizing the processor with minimum clock period = 7ns and the timing constrains are MET
- Synopsys results an area = 126528