Computer Architecture Lab 2

Name: Eddy Kimani

Reg No.: SCT212-0596/2021

E1: Fetch Decode Execute

1. Without Forwarding and branch optimization

Instr	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
LD	F	D	X	M	W															
DADDI		F	D	s	s	X	M	W												
SD			F	s	s	D	s	s	X	M	W									
DADDI						F	s	s	D	X	M	W								
DSUB									F	D	s	s	X	M	W					
BNEZ										F	s	s	D	s	s	X	M	W		
LD 2													F	s	s	F	D	X	M	W

- The first DADDI instruction must wait until the LD instruction reaches the WB stage to get the value of R1.
- The SD instruction must wait for the first DADDI to compute R1 and reach the WB stage, since there is no forwarding.
- The DSUB instruction must wait until the second DADDI computes R2 and reaches the WB stage.
- The BNEZ instruction must wait for DSUB to compute R4 and reach the WB stage.
- The LD instruction from the next iteration is re-fetched only after the branch is resolved as taken in the ID stage.

2. Forwarding and branch predict not taken

Instr	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LD	F	D	X	M	W									
DADDI		F	D	s	X	M	W							
SD			F	s	D	X	M	W						
DADDI					F	D	X	M	W					
DSUB						F	D	X	M	W				
BNEZ							F	D	s	X	M	W		
LD 2								F	s	F	D	X	M	W

- The first ADDI still has to wait until the LD instruction reaches the WB stage to get the value of R1, but now the stall is handled in the EX stage using interlocking, and the value is forwarded directly to the EX stage.
- The SD instruction now waits for the first ADDI to compute R1 and forward it from the EX stage to the MEM stage.
- The DSUB instruction now receives the value of R2 directly via forwarding from the second ADDI, so it doesn't need to wait.
- The BNEZ instruction gets the value of R4 forwarded from the SUB instruction, but it must wait until SUB reaches the EX stage.
- The LD instruction from the next iteration is re-fetched after the branch is resolved as mispredicted during the ID stage.

3. Instruction scheduling and branch delay slot

Instr	1	2	3	4	5	6	7	8	9	10	11
LD R1, 0(R2)	F	D	X	M	W						
DADDI R2, R2, 4		F	D	X	M	W					
DSUB R4, R3, R2			F	D	X	M	W				
DADDI R1, R1, 1				F	D	X	M	W			
BNEZ R4, loop					F	D	X	M	W		
SD -4(R2), R1						F	D	X	M	W	
LD (2)							F	D	X	M	W

- Optimization 1: The second DADDI is moved into the load delay slot.
- Optimization 2: The second DSUB is moved up to follow the second DADDI.
- Optimization 3: The SD instruction is moved into the branch delay slot, with its offset adjusted to SD -4(R2), R1.

- The first DADDI no longer has to wait, since it is now one cycle apart from the LD instruction.
- The BNEZ instruction no longer waits, since it is now separated by an extra cycle from the DSUB.
- The SD instruction now occupies the branch delay slot.
- The LD instruction from the next iteration is fetched correctly, as the branch is resolved in time.