Left neighbour	Your Name	Your Entry No.	Right neighbour

COL216 Computer Architecture Quiz 2 (Set A) 16.03.2017

Q 1. While carrying out a multi-cycle design for ARM processor, it is required to figure out how various instructions will be broken up into cycles. Show how will you break up the instructions given below into cycles, considering the following datapath constraints - (a) the register file has only 2 read ports and 1 write port, (b) the ALU is used to do all arithmetic and logical operations as well as address calculations (there are no separate adders), (c) multiplication and shifting/rotation cannot be done in the same cycle in which ALU is being used. Assume that PC is a separate register, rather than R15.

LDREQ Rd, [Rn, - Rm, LSL #2] ! {Load if equal, with auto (pre) decrement}
ANDS Rd, Rn, Rm, LSR Rs {Perform bit-wise AND and set flags}
MUL Rd, Rm, Rs {Multiply Rd = Rm * Rs}

Steps	LDREQ Rd, [Rn, - Rm, LSL #2] !	ANDS Rd, Rn, Rm, LSR Rs	MUL Rd, Rm, Rs
1	IR = Mem[PC] PC = PC+4	IR = Mem[PC] PC = PC+4	IR = Mem[PC] PC = PC+4
2	A = RF[IR[19-16]] B = RF[IR[3-0]]	A = RF[IR[19-16]] B = RF[IR[3-0]]	A = RF[IR[19-16]] B = RF[IR[3-0]]
3	D = shift(B, LSL, 2)	X = RF[IR[11-8]]	X = RF[IR[11-8]]
4	Res = ALU(A, D, -)	D = shift(B, LSR, X[4-0])	Res = X * B
5	DR = Mem[Res] if (Z) RF[IR[19-16]] = Res	Res = ALU(A, D, and) Set flags	RF[IR[19-16]] = Res
6	if (Z) RF[IR[15-12]] = DR	RF[IR[15-12]] = Res	

[1 mark] [1 mark] [1 mark]

Q 2. Consider a classical 5-stage pipeline to implement ARM instructions DP, DT and B. For DP instructions, no shift/rotate is allowed. For DT instructions, offset is always 0. Writing into Register file takes place at the end of a clock cycle. When a branch instruction is detected, any instructions entered in pipeline after the branch instructions are flushed. When the branch completes the 3rd stage, its correct successor is fetched. Find the number of clock cycles required to execute the code below in two cases - (a) there are no data forwarding paths and (b) all possible data forwarding paths are there. Show how you obtain your answer. Assume that the loop executes 10 times.

```
Loop:  LDR R4, [R1]
       LDR R5, [R2]
       ADD R4, R4, R5
       STR R4, [R1]
       ADD R1, R1, #4
       ADD R2, R2, #4
       CMP R1, R6
       BLT  Loop
```

Instruction	start cycle, no forwarding	start cycle, with forwarding
Loop: LDR R4, [R1]	1	1
LDR R5, [R2]	2	2
ADD R4, R4, R5	6	4
STR R4, [R1]	10	5
ADD R1, R1, #4	11	6
ADD R2, R2, #4	12	7
CMP R1, R6	15	8
BLT Loop	16	9
next iteration starts at	19	12

A data dependent instruction is delayed by 3 cycles. This is reduced to 0 with data forwarding except for the case of LDR followed by DP where it is reduced to 1. A control dependent instruction is delayed by 2 cycles.

Without forwarding, cycles per iteration = 18, total cycles for 10 iteration = 180. [1 mark]

Without forwarding, cycles per iteration = 11, total cycles for 10 iteration = 110. [1 mark]

Left neighbour	Your Name	Your Entry No.	Right neighbour

COL216 Computer Architecture Quiz 2 (Set B) 16.03.2017

Q 1. While carrying out a multi-cycle design for ARM processor, it is required to figure out how various instructions will be broken up into cycles. Show how will you break up the instructions given below into cycles, considering the following datapath constraints - (a) the register file has only 2 read ports and 1 write port, (b) the ALU is used to do all arithmetic and logical operations as well as address calculations (there are no separate adders), (c) multiplication and shifting/rotation cannot be done in the same cycle in which ALU is being used. Assume that PC is a separate register, rather than R15.

LDRGT Rd, [Rn], - Rm, LSL #2 {Load if greater, with auto (post) decrement}
ORRS Rd, Rn, Rm, LSR #4 {Perform bit-wise OR and set flags}
MLA Rd, Rm, Rs, Rn {Multiply accumulate Rd = Rm * Rs + Rn}

Steps	LDRGT Rd, [Rn], - Rm, LSL #2	ORRS Rd, Rn, Rm, LSR #4	MLA Rd, Rm, Rs, Rn
1	IR = Mem[PC] PC = PC+4	IR = Mem[PC] PC = PC+4	IR = Mem[PC] PC = PC+4
2	A = RF[IR[19-16]] B = RF[IR[3-0]]	A = RF[IR[19-16]] B = RF[IR[3-0]]	A = RF[IR[19-16]] B = RF[IR[3-0]]
3	D = shift(B, LSL, 2)	D = shift(B, LSR, 4)	A = RF[IR[15-11]] X = RF[IR[11-8]]
4	Res = ALU(A, D, -)	Res = ALU(A, D, or), Set flags	D = X * B
5	DR = Mem[A] if(Z' and N=V)RF[IR[19-16]] = Res	RF[IR[15-12]] = Res	Res = ALU(A, D, +)
6	if(Z' and N=V)RF[IR[15-12]] = DR		RF[IR[19-16]] = Res
	[1 mark]	[1 mark]	[1 mark]

Q 2. Consider a classical 5-stage pipeline to implement ARM instructions DP, DT and B. For DP instructions, no shift/rotate is allowed. For DT instructions, offset is always 0. Writing into Register file takes place at the end of a clock cycle. When a branch instruction is detected, any instructions entered in pipeline after the branch instructions are flushed. When the branch completes the 3rd stage, its correct successor is fetched. Find the number of clock cycles required to execute the code below in two cases - (a) there are no data forwarding paths and (b) all possible data forwarding paths are there. Show how you obtain your answer. Assume that the loop executes 10 times.

```
Loop: LDR R4, [R1]
      CMP R4, R5
      ADDEQ R3, R3, #1
      STR R4, [R2]
      ADD R1, R1, #4
      ADD R2, R2, #4
      CMP R1, R6
      BLT Loop
```

Instruction	start cycle, no forwarding	start cycle, with forwarding
Loop: LDR R4, [R1]	1	1
CMP R4, R5	5	3
ADDEQ R3, R3, #1	6	4
STR R4, [R2]	7	5
ADD R1, R1, #4	8	6
ADD R2, R2, #4	9	7
CMP R1, R6	12	8
BLT Loop	13	9
next iteration starts at	16	12

A data dependent instruction is delayed by 3 cycles. This is reduced to 0 with data forwarding except for the case of LDR followed by DP where it is reduced to 1. A control dependent instruction is delayed by 2 cycles.

Without forwarding, cycles per iteration = 15, total cycles for 10 iteration = 150. [1 mark]

Without forwarding, cycles per iteration = 11, total cycles for 10 iteration = 110 [1 mark]