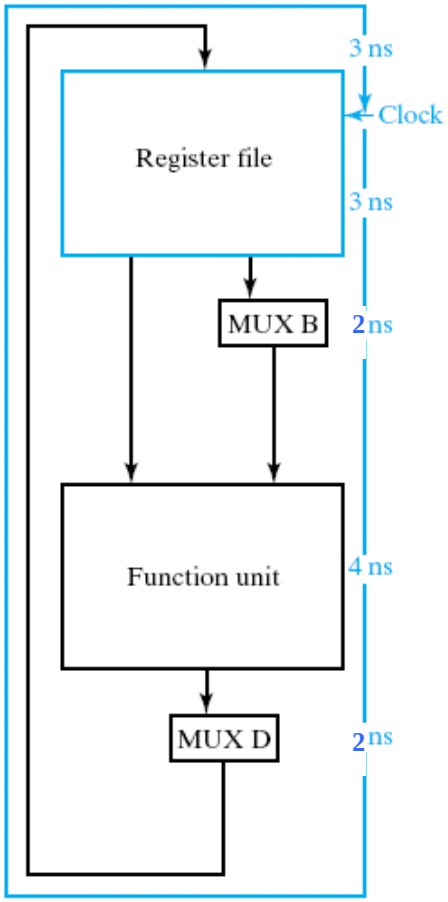
Tutorial 9 – Sample Solutions
RISC and CISC

CO 2206 Computer Organization

Task 1. Single-Cycle Computer Clock Limit



Processing of a microoperation involves fetching the operand(s) from the register file and appropriately select the multiplex MUX B, selecting and performing the requested operation on the operand(s), appropriately selecting the result/data at the mulitiplexer MUX D and finally writing the result/data back to the register. These events process the microoperation through the datapath. Figure to the left shows the datapath of an example CPU. If the times required to pass through the different circuitry in the datapath is as shown in the figure, determine the maximum clock frequency that can be used with this CPU.

Task 1. Ans.

- Minimum clock period = time to pass through the datapath =3+3+2+4+2 =14 ns
- Max clock frequency = 1/ (min clock period) = **71.43 MHz**

Task 2. Pipelining

- What is the purpose of using pipelining?
- Define latency and instruction throughput?
- Will pipelining decrease or increase latency time?
- Will pipelining decrease or increase instruction throughput?
- Determine the latency and instruction throughput when executing 3 instructions in a computer using 4-stage pipelining and a clock speed of 500MHz.

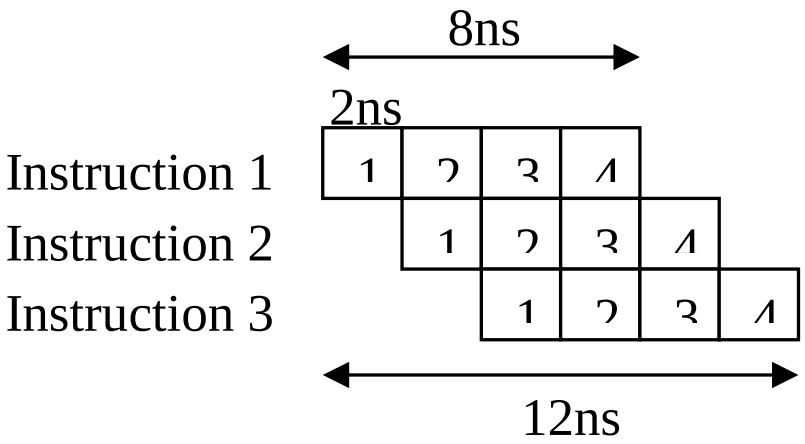
Task 2. Ans (1)

- To improve CPU performance by improving instruction throughput. This is achieved through exploiting parallelism within the circuitry of the CPU.
- Instruction Latency is the time required to process one instruction (from fetch to execute). Instruction Throughput is the number of instructions that can be processed per unit time (second).
- Increase
- Increase
- Referring to the diagram below,

Task 2. Ans (2)

Time for each stage = clock period = 1/frequency = 1 /500M =2ns

Latency = no. of stages x clock period = 4x2ns =8ns



Throughput = no. of instructions / total time = 3 / 12ns =250 MIPS

(MIPS = Mega Instructions Per Second)

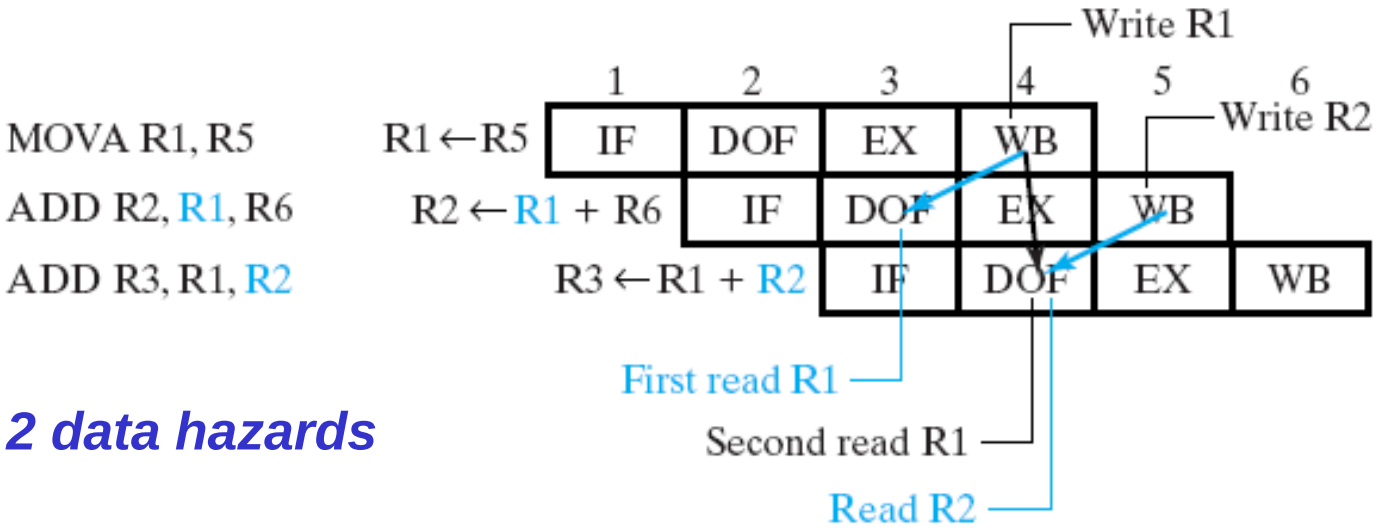
Task 3. Hazards in Pipelining

- What is data hazard in a pipelined datapath? Illustrate your explanation using appropriate example.
- Using appropriate illustrations, explain how data forwarding overcome data hazards.

Task 3. Ans (1)

- Data hazard is a timing problem in pipelining, which occurs if a subsequent instruction tries to use the result of an operation as an operand before the result is available. The diagram below (next slide) shows 2 data hazards in the pipeline. The second instruction reads an operand (R1) that is being modified by first instruction, however before it is being written back (WB). Similarly, third the third instruction reads an operand (R2) that is being modified by the second instruction.

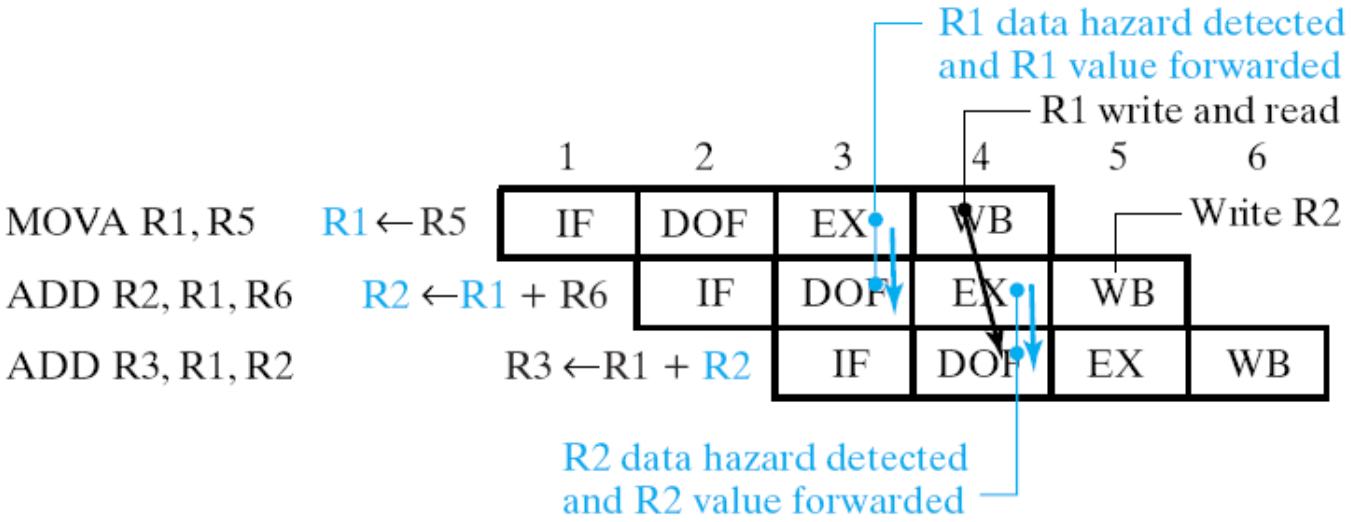
Task 3. Ans (2)



2 data hazards

Task 3. Ans (3)

- Data forwarding uses a circuitry to detect data hazard by looking at control signals at DOF and EX stage. When a data hazard is detected, the circuit route the result from EX stage (before it is being written back) to the DOF as operand.



Task 4. RISC

- Referring to the RISC computer described in the lecture, answer the following questions:
  - What does it mean by hardwired control unit? What would be used if it the control unit was not hardwired?
  - Why does the processor require increased amount of registers, as compared to the single-cycle computer described in Chapter 8?
  - What is the maximum number of distinct instructions that can be specified? Explain how this number is derived.
  - How many bits does the immediate (constant) field has? Why do we need to perform zero fill or sign extend on the constant?
  - Referring to the instruction set given in the lecture slide, give one example for each of the four addressing modes below:
    - Register
    - Register Indirect
    - Immediate
    - Relative

Task 4. Ans (1)

- Hardwired control unit has the instruction decoding fixed through logic circuit, whereby each instruction will be directly decoded into a control word. Alternative to hardwired control unit will be the microcode, whereby each instruction will be decoded into a sequence of microcodes (microcode programming) which in turn will be converted to corresponding control words.
- The RISC computer has simple instructions, which only perform operations on registers (apart from the two Load and Save instructions). To minimize memory access (hence improve performance), large amount of registers is required to store temporary data during instruction processing (and along the pipeline). The larger size (width) of the registers also allows one single instruction to hold operands (and addresses).

Task 4. Ans (2)

- 7-bit (A31||A25) opcode field giving a maximum of 27 =128 distinct instructions
- Immediate field has 15 bits (A14||A0). The immediate value is 15-bit while registers are 32-bit. To store the immediate value supplied in the instruction into specified register, the immediate value has to be extended to 32-bit. For non-arithmetic operations, to extend the immediate value to 32-bit simply requires filling the higher order bits (A32||A15) with zeros, i.e. zero fill. For arithmetic operations, the sign of the immediate value need to be taken into consideration by duplicating the MSB (A14) of the immediate field into the higher order bits (A32||A15), i.e. sign extend.
  - Register AM: MOVA, ADD, SUB, AND, OR, …
  - Register Indirect AM: Load, Store
  - Immediate AM: ADI, SBI, ANI, ORI, …
  - Relative AM: BZ, BNZ, JMP, JML