MODULE 5 important questions

1 **With a figure, explain single bus organization of datapath inside a processor.**

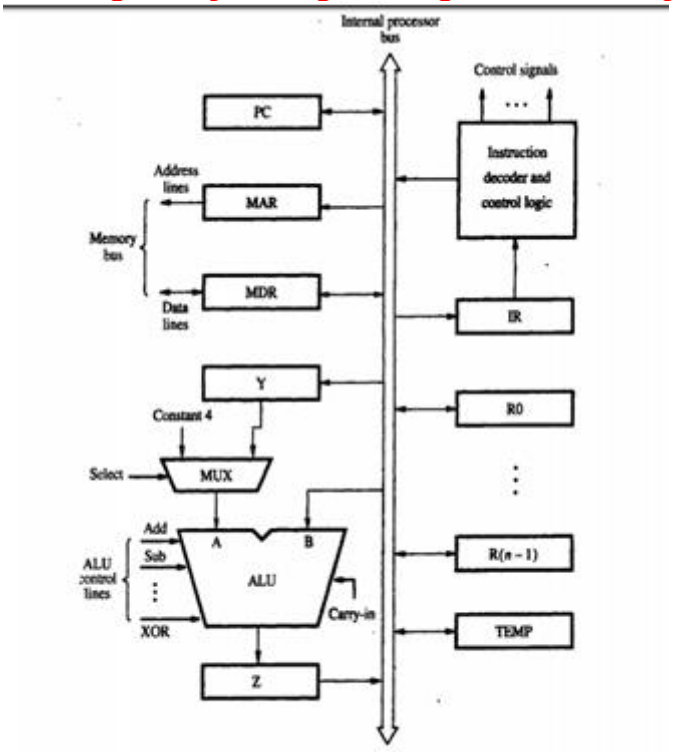


Figure 7.1 Single-bus organization of the datapath inside a processor.

SINGLE BUS ORGANIZATION

- ALU and all the registers are interconnected via a **Single Common Bus** (Figure 7.1).
- Data & address lines of the external memory-bus is connected to the internal processor-bus via MDR & MAR respectively. (MDR→ Memory Data Register, MAR → Memory Address Register).
- **MDR** has 2 inputs and 2 outputs. Data may be loaded
  - → into MDR either from memory-bus (external) or
  - → from processor-bus (internal).
- **MAR**'s input is connected to internal-bus;
  MAR's output is connected to external-bus.
- **Instruction Decoder & Control Unit** is responsible for
  - → issuing the control-signals to all the units inside the processor.
  - → implementing the actions specified by the instruction (loaded in the IR).
- Register R0 through R(n-1) are the **Processor Registers**.
  The programmer can access these registers for general-purpose use.
- Only processor can access 3 registers **Y**, **Z** & **Temp** for temporary storage during program-execution.
  The programmer cannot access these 3 registers.
- In **ALU**, 1) 'A' input gets the operand from the output of the multiplexer (MUX).
  2) 'B' input gets the operand directly from the processor-bus.
- There are 2 options provided for 'A' input of the ALU.
- MUX is used to select one of the 2 inputs.
- **MUX** selects either
  - → output of Y or
  - → constant-value 4(which is used to increment PC content).

- An instruction is executed by performing one or more of the following operations:
  1) Transfer a word of data from one register to another or to the ALU.
  2) Perform arithmetic or a logic operation and store the result in a register.
  3) Fetch the contents of a given memory-location and load them into a register.
  4) Store a word of data from a register into a given memory-location.
- **Disadvantage:** Only one data-word can be transferred over the bus in a clock cycle.
  Solution: Provide multiple internal-paths. Multiple paths allow several data-transfers to take place in parallel.

2.Give the control sequence for execution of complete instruction ADD (R3), R1.

Add (R3), R1

Step	Action
1	PC_{out} , MAR_{in} , Read, Select4,Add, Z_{in}
2	Z_{out} , PC_{in} , Y_{in} , WMF C
3	MDR_{out} , IR_{in}
4	$R3_{out}$, MAR_{in} , Read
5	$R1_{out}$, Y_{in} , WMF C
6	MDR_{out} , SelectY, Add, Z_{in}
7	Z_{out} , $R1_{in}$, End

- Instruction execution proceeds as follows:

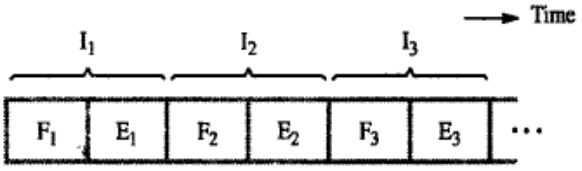
Step1--> The instruction-fetch operation is initiated by
→ loading contents of PC into MAR &
→ sending a Read request to memory.
The Select signal is set to Select4, which causes the Mux to select constant 4. This value is added to operand at input B (PC's content), and the result is stored in Z.
Step2--> Updated value in Z is moved to PC. This completes the PC increment operation and PC will now point to next instruction.
Step3--> Fetched instruction is moved into MDR and then to IR.
The step 1 through 3 constitutes the **Fetch Phase**.
At the beginning of step 4, the instruction decoder interprets the contents of the IR. This enables the control circuitry to activate the control-signals for steps 4 through 7.
The step 4 through 7 constitutes the **Execution Phase**.
Step4--> Contents of R3 are loaded into MAR & a memory read signal is issued.
Step5--> Contents of R1 are transferred to Y to prepare for addition.
Step6--> When Read operation is completed, memory-operand is available in MDR, and the addition is performed.
Step7--> Sum is stored in Z, then transferred to R1.The End signal causes a new instruction fetch cycle to begin by returning to step1.

3 Explain the differences between Hardwired and Micro-programmed control.

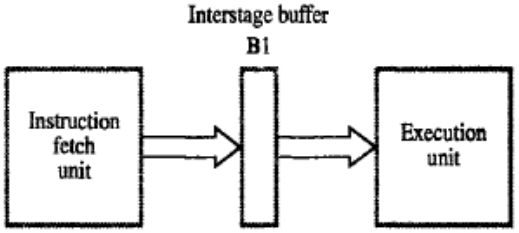
Attribute	Hardwired Control	Microprogrammed Control
Definition	Hardwired control is a control mechanism to generate control-signals by using gates, flip-flops, decoders, and other digital circuits.	Micro programmed control is a control mechanism to generate control-signals by using a memory called control store (CS), which contains the control-signals.
Speed	Fast	Slow
Control functions	Implemented in hardware.	Implemented in software.
Flexibility	Not flexible to accommodate new system specifications or new instructions.	More flexible, to accommodate new system specification or new instructions redesign is required.
Ability to handle large or complex instruction sets	Difficult.	Easier.
Ability to support operating systems & diagnostic features	Very difficult.	Easy.
Design process	Complicated.	Orderly and systematic.
Applications	Mostly RISC microprocessors.	Mainframes, some microprocessors.
Instructionset size	Usually under 100 instructions.	Usually over 100 instructions.
ROM size	-	2K to 10K by 20-400 bit microinstructions.
Chip area efficiency	Uses least area.	Uses more area.

4.. **Explain the basic idea of instruction pipelining operation ?**

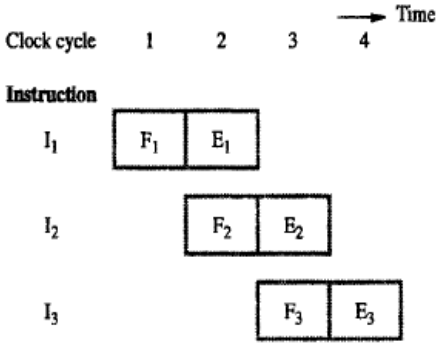
The speed of execution of programs is influenced by many factors. One way to improve performance is to use faster circuit technology to build the processor and the main memory. Another possibility is to arrange the hardware so that more than one operation can be performed at the same time. In this way, the number of operations performed per second is increased even though the elapsed time needed to perform any one operation is not changed.



(a) Sequential execution



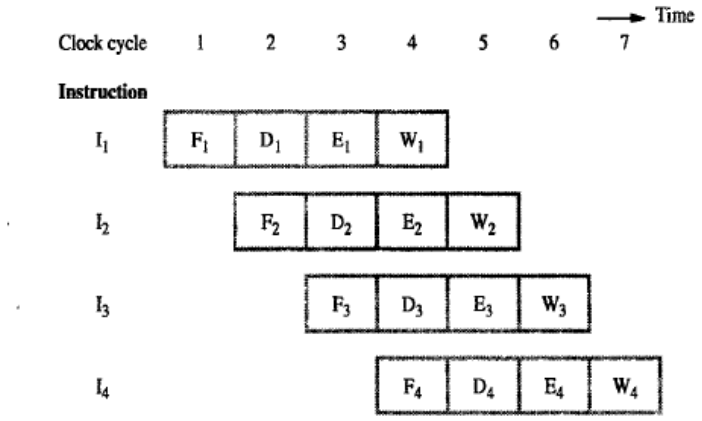
(b) Hardware organization



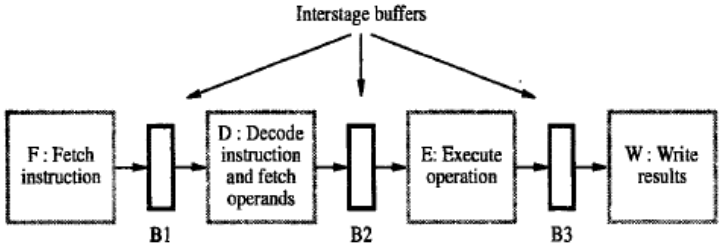
(c) Pipelined execution

Figure 8.1 Basic idea of instruction pipelining.

The computer is controlled by a clock whose period is such that the fetch and execute steps of any instruction can each be completed in one clock cycle. Operation of the computer proceeds as in Figure 8.1c. In the first clock cycle, the fetch unit fetches an instruction I_1 (step F_1) and stores it in buffer B1 at the end of the clock cycle. In the second clock cycle, the instruction fetch unit proceeds with the fetch operation for instruction I_2 (step F_2). Meanwhile, the execution unit performs the operation specified by instruction I_1, which is available to it in buffer B1 (step E_1). By the end of the



(a) Instruction execution divided into four steps



(b) Hardware organization

5. Explain ORGANIZATION OF MICROPROGRAMMED CONTROL UNIT (TO SUPPORT CONDITIONAL BRANCHING)

• In case of conditional branching, microinstructions specify which of the external inputs, condition-codes should be checked as a condition for branching to take place.
• The *starting and branch address generator block* loads a new address into μPC when a microinstruction instructs it to do so.
• To allow implementation of a conditional branch, inputs to this block consist of
  → external inputs and condition-codes
  → contents of IR
• μPC is incremented every time a new microinstruction is fetched from microprogram memory except in following situations
  i) When a new instruction is loaded into IR, μPC is loaded with starting-address of microroutine for that instruction.
  ii) When a Branch microinstruction is encountered and branch condition is satisfied, μPC is loaded with

branch-address.

iii) When an End microinstruction is encountered, µPC is loaded with address of first CW in microroutine for instruction fetch cycle.

Address	Microinstruction
0	PC_{out}, MAR_{in}, Read, Select4, Add, Z_{in}
1	Z_{out}, PC_{in}, Y_{in}, WMFC
2	MDR_{out}, IR_{in}
3	Branch to starting address of appropriate microroutine
...	...
25	If N=0, then branch to microinstruction 0
26	Offset-field-of-IR_{out}, SelectY, Add, Z_{in}
27	Z_{out}, PC_{in}, End

Figure 7.17 Microroutine for the instruction Branch < 0.

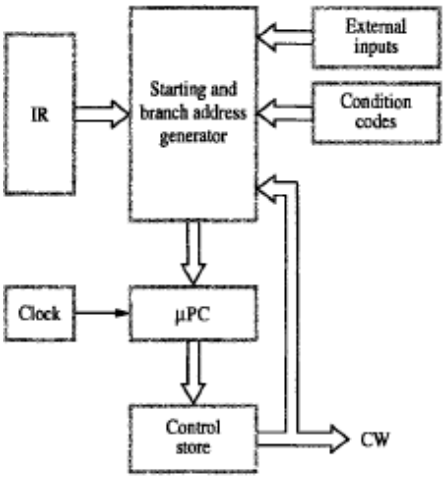


Figure 7.18 Organization of the control unit to allow conditional branching in the microprogram.

6. Explain HARDWIRED CONTROL

• Decoder/encoder block is a combinational-circuit that generates required control-outputs depending on state of all its inputs.

• Step-decoder provides a separate signal line for each step in the control sequence.
Similarly, output of instruction-decoder consists of a separate line for each machine instruction.

• For any instruction loaded in IR, one of the output-lines INS_1 through INS_m is set to 1, and all other lines are set to 0.

• The input signals to encoder-block are combined to generate the individual control-signals Y_{in}, PC_{out}, Add, End and so on.

• For example, $Z_{in}=T_1+T_6.ADD+T_4.BR$;This signal is asserted during time-slot T_1 for all instructions,
during T_6 for an Add instruction
during T_4 for unconditional branch instruction

• When RUN=1, counter is incremented by 1 at the end of every clock cycle.
When RUN=0, counter stops counting.

• Sequence of operations carried out by this machine is determined by wiring of logic elements, hence the name "hardwired".

• Advantage: Can operate at high speed.
Disadvantage: Limited flexibility.

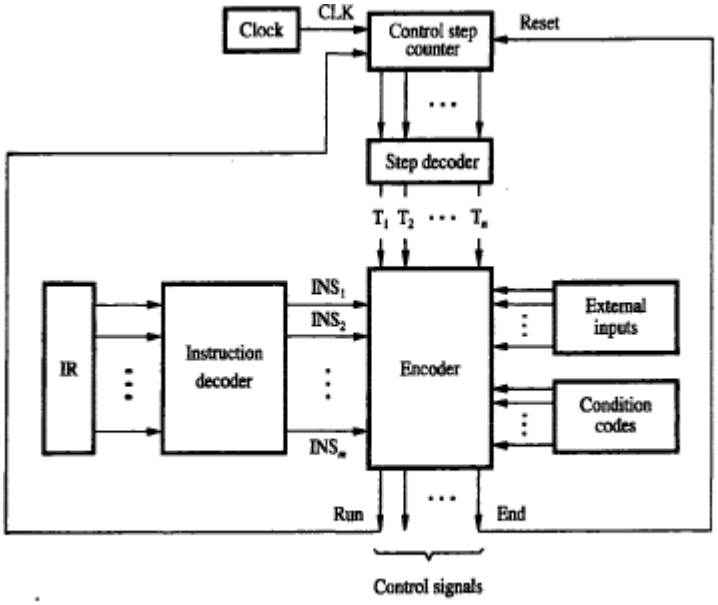


Figure 7.11 Separation of the decoding and encoding functions.

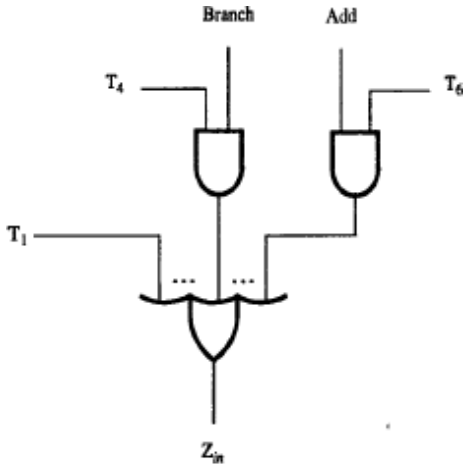


Figure 7.12 Generation of the Z_{in} control signal for the processor in Figure 7.1.

7. Explain MULTIPLE BUS ORGANIZATION

- All general-purpose registers are combined into a single block called the *register file*.
- Register-file has 3 ports. There are 2 outputs allowing the contents of 2 different registers to be simultaneously placed on the buses A and B.
- Register-file has 3 ports.
    1) Two output-ports allow the contents of 2 different registers to be simultaneously placed on buses A & B.
    2) Third input-port allows data on bus C to be loaded into a third register during the same clock-cycle.
- Buses A and B are used to transfer source-operands to A & B inputs of ALU.
- Result is transferred to destination over bus C.
- Incrementer-unit is used to increment PC by 4.
- Control sequence for the instruction *Add R4,R5,R6* is as follows
    1) PC_{out}, R=B, MAR_{in}, Read, IncPC
    2) WMFC
    3) MDR_{out}, R=B, IR_{in}
    4) $R4_{outA}$, $R5_{outB}$, SelectA, Add, $R6_{in}$, End
- Instruction execution proceeds as follows:

    Step 1--> Contents of PC are passed through ALU using R=B control-signal and loaded into MAR to start a memory Read operation. At the same time, PC is incremented by 4.

    Step2--> Processor waits for MFC signal from memory.

    Step3--> Processor loads requested-data into MDR, and then transfers them to IR. Step4--> The instruction is decoded and add operation take place in a single step.

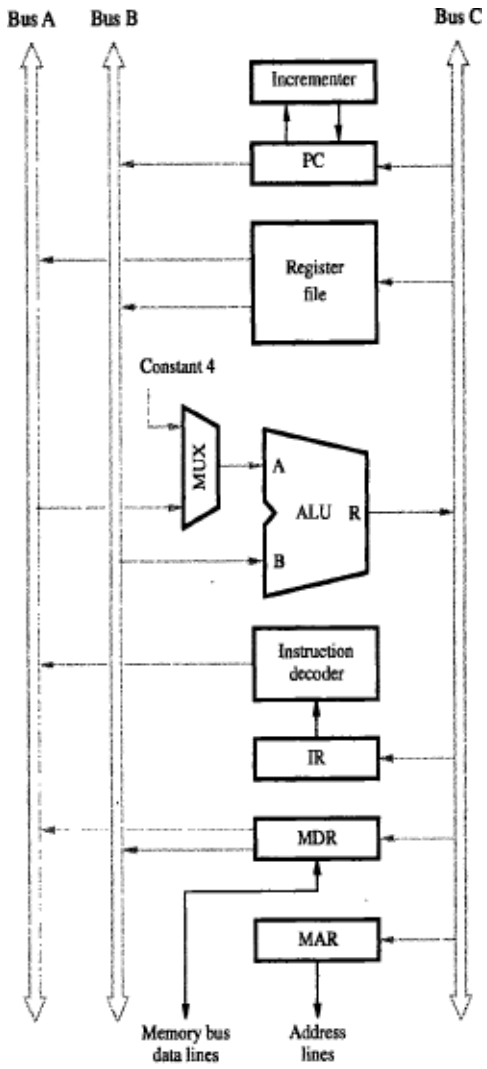


Figure 7.8 Three-bus organization of the datapath.

Note: