SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

AN AUTONOMOUS INSTITUTION

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

PUZZLES

UNIT 4- PROCESSOR

Puzzle 1: Fetch-Execute Cycle

Question: Describe the fetch-execute cycle and identify the components involved at each step.

Answer: The fetch-execute cycle consists of:

1. **Fetch**: Retrieve instruction from memory (PC, MAR, IR).
2. **Decode**: Interpret the instruction (Control Unit).
3. **Execute**: Perform the operation (ALU, Registers).
4. **Store**: Write back results (Registers, Memory).

Puzzle 2: Opcode Challenge

Question: Given the following opcodes, categorize them into arithmetic, logic, and control instructions: `ADD, SUB, AND, OR, JMP`.

Answer:

- **Arithmetic**: ADD, SUB
- **Logic**: AND, OR
- **Control**: JMP

Puzzle 3: Data Path Layout

Question: Draw a data path for a simple ALU operation (e.g., ADD) and label each component.

Answer: Diagram includes:

- **Registers**: R1, R2 (inputs), R3 (output)
- **ALU**: Performs ADD
- **Multiplexer**: Selects inputs for the ALU

Puzzle 4: Signal Routing

Question: Given a data path diagram, identify the correct paths for executing an ADD instruction.

Answer: The paths should route data from R1 and R2 through the ALU to R3.

Puzzle 5: Control Signals

Question: Create a truth table for control signals for the instructions `ADD` and `SUB`.

Answer:

Instruction	ALUop	RegWrite	MemRead	MemWrite
ADD	0	1	0	0
SUB	1	1	0	0

Puzzle 6: Hardwired vs. Microprogrammed

Question: Compare hardwired control and microprogrammed control using examples.

Answer:

- **Hardwired Control**: Fast, uses fixed logic circuits (e.g., simple CPUs).
- **Microprogrammed Control**: Flexible, uses stored programs for control signals (e.g., complex instruction sets).

Puzzle 7: Circuit Design

Question: Design a simple hardwired control unit for a CPU that supports ADD and SUB.

Answer: Use logic gates to generate control signals based on opcode inputs.

Puzzle 8: Timing Diagram

Question: Draw a timing diagram for a hardwired control unit executing two instructions.

Answer: Diagram shows control signals changing states synchronized with clock cycles for ADD and SUB.

Puzzle 9: Microinstruction

Question: Given a set of microinstructions, sequence them to perform a load operation.

Answer:

1. Fetch the instruction.
2. Decode the instruction.
3. Access memory.
4. Write result to register.

Puzzle 10: Microprogram Control

Question: Describe how a microprogrammed control unit can accommodate a new instruction.

Answer: A new instruction can be added by defining its microinstruction sequence in the control store.

Puzzle 11: Pipeline Stages

Question: List and describe the stages of instruction pipelining.

Answer:

1. **IF**: Instruction Fetch
2. **ID**: Instruction Decode
3. **EX**: Execute
4. **MEM**: Memory Access
5. **WB**: Write Back

Puzzle 12: Throughput Calculation

Question: Calculate the throughput of a pipeline with five stages.

Answer: Ideal throughput is 5 instructions every 5 cycles, or 1 instruction per cycle.

Puzzle 13: Identifying Hazards

Question: Identify potential data hazards in the following sequence:

```
markdown
Copy code
1. R1 = R2 + R3
2. R4 = R1 + R5
```

Answer: There is a RAW (Read After Write) hazard on R1.

Puzzle 14: Hazard Resolution

Question: Propose a strategy to resolve a specific data hazard in a pipelined architecture.

Answer: Use data forwarding to send the result of R1 directly to the next instruction instead of waiting for it to be written back.

Puzzle 15: Branch Prediction

Question: Explain how branch prediction works and its impact on pipeline efficiency.

Answer: Branch prediction guesses whether a branch will be taken, reducing stalls and improving pipeline efficiency by allowing speculative execution.

Puzzle 16: Control Hazard Example

Question: Given a set of branch instructions, determine the number of stalls in a pipeline.

Answer: If a branch is mispredicted, it can introduce 2 stalls until the correct path is resolved.

Puzzle 17: CPU Design

Question: Sketch a simple CPU architecture and label its main components.

Answer: Components include ALU, Registers, Control Unit, Memory, and data/address buses.

Puzzle 18: Instruction Set Classification

Question: Classify the following instruction set into RISC or CISC: `ADD, SUB, JMP, MOV`.

Answer: This instruction set is typically RISC due to its simple operations and fixed length.

Puzzle 19: Performance Metrics

Question: Define and calculate latency and throughput for a CPU.

Answer:

- **Latency**: Time to complete one operation.
- **Throughput**: Number of operations completed in a given time (e.g., 100 operations in 10 seconds = 10 ops/sec).

Puzzle 20: Memory Hierarchy

Question: Explain how the memory hierarchy affects CPU performance.

Answer: The hierarchy (registers, cache, RAM, disk) balances speed and size, with faster, smaller memories improving access time, while larger memories provide more storage.

Puzzle 21: Design a Control Unit

Question: Outline control signals for operations: `ADD`, `SUB`, `LOAD`, `STORE`.

Answer: Define control signals for each operation indicating ALU operations, memory access, and register writes.

Puzzle 22: Pipeline Conflict Resolution

Question: Propose a conflict resolution strategy for a pipeline processing multiple instruction types.

Answer: Implement forwarding and hazard detection units to handle data hazards, and insert stalls when necessary.

Puzzle 23: Logic Circuit

Question: Create a logic circuit that outputs a signal when two specific control signals are high.

Answer: Use an AND gate to combine the two control signals; the output will be high when both inputs are high.

Puzzle 24: State Machine Design

Question: Design a simple state machine for a control unit during instruction execution.

Answer: States include Fetch, Decode, Execute, Write Back, with transitions based on current instruction.

Puzzle 25: Case Study Analysis

Question: Analyze a real-world CPU architecture focusing on its control unit design.

Answer: For example, ARM architecture uses a complex instruction set with a microprogrammed control unit, leveraging pipelining and out-of-order execution for efficiency.