|  |  |
| --- | --- |
| Surname & Initials | Student Number |
| Monokoa T.J | 201600428 |
| Mantsi R | 202201932 |
| Mokotoi T.M | 202321234 |
| Kobeli | 202322593 |
| Tjabafu | 202322637 |
| Mapola | 202322602 |
| Mokotjo R.M | 202321189 |

**Microarchitecture Specification - Custom AI Processor**

**1. Pipeline Overview**

**5-Stage Pipelined Datapath: IF → ID → EX → MEM → WB**

Clock Cycle: 1 2 3 4 5 6 7

Instruction 1: IF → ID → EX → MEM → WB

Instruction 2: IF → ID → EX → MEM → WB

Instruction 3: IF → ID → EX → MEM → WB

**2. Complete Datapath Components**

**IF Stage (Instruction Fetch)**

* **Program Counter (PC)**: 32-bit register
* **Instruction Memory**: 16-bit instructions, 32-bit addresses
* **PC Incrementer**: PC + 2 (16-bit instructions)
* **PC MUX**: Selects next PC (sequential/branch)
* **IF/ID Pipeline Register**: Stores instruction + PC+2

**ID Stage (Instruction Decode)**

* **Control Unit**: Generates all control signals
* **Register File**: 16 registers (x0-x15), 32-bit data
* **Sign Extension Unit**: 4-bit → 32-bit immediate
* **ID/EX Pipeline Register**: Stores control + data

**EX Stage (Execute)**

* **ALU**: Arithmetic/logic operations
* **MUX A**: Selects ALU input A (Read1/PC+2)
* **MUX B**: Selects ALU input B (Read2/Immediate/Constant)
* **Custom AI Units**:
  + **VCMPEQ.B**: Vector byte comparison
  + **BCNT**: Bit count for biometrics
  + **MAC**: Multiply-accumulate for neural networks
  + **SLEEPM**: Power control
* **Result MUX**: Selects output (ALU/custom units)
* **EX/MEM Pipeline Register**: Stores results + control

**MEM Stage (Memory Access)**

* **Data Memory**: 32-bit read/write
* **MEM/WB Pipeline Register**: Stores memory data + results

**WB Stage (Write Back)**

* **MUX C**: Selects write data (ALU/Memory/PC+2)
* **Write to Register File**

**3. Control Signals**

|  |  |  |
| --- | --- | --- |
| **Signal** | **Width** | **Function** |
| RegWrite | 1-bit | Enable register write |
| MemRead | 1-bit | Enable memory read |
| MemWrite | 1-bit | Enable memory write |
| MemToReg | 2-bit | WB MUX select (00=ALU, 01=Mem, 10=PC+2) |
| ALUSrcA | 1-bit | MUX A select (0=Read1, 1=PC+2) |
| ALUSrcB | 2-bit | MUX B select (00=Read2, 01=Imm, 10=Const) |
| ALUOp | 4-bit | ALU operation code |
| PCSrc | 1-bit | PC MUX select (0=PC+2, 1=Branch) |
| VCMPEnable | 1-bit | Enable vector compare |
| POPCNTEnable | 1-bit | Enable bit count |
| MACEnable | 1-bit | Enable multiply-accumulate |
| SLEEPEnable | 1-bit | Enable sleep mode |

**4. Hazard Handling**

**Forwarding Unit**

* Detects RAW (Read-After-Write) hazards
* Forwards data from EX/MEM and MEM/WB stages
* **ForwardA/ForwardB**: 2-bit control for forwarding MUXes

**Forwarding Logic**

text

if (EX/MEM.RegWrite and EX/MEM.rd == ID/EX.rs1)

ForwardA = 01 (forward from EX/MEM)

else if (MEM/WB.RegWrite and MEM/WB.rd == ID/EX.rs1)

ForwardA = 10 (forward from MEM/WB)

else

ForwardA = 00 (normal)

**5. Custom AI Units Specification**

**VCMPEQ.B (Vector Compare)**

* **Input**: Two 32-bit values
* **Output**: 32-bit comparison mask
* **Function**: Parallel byte-wise comparison
* **Use Case**: Voice keyword matching

**BCNT (Bit Count)**

* **Input**: 32-bit value
* **Output**: 5-bit count (0-32)
* **Function**: Population count
* **Use Case**: Biometric feature matching

**MAC (Multiply-Accumulate)**

* **Input**: Two 32-bit values + accumulator
* **Output**: 32-bit result
* **Function**: result = accumulator + (A × B)
* **Use Case**: Neural network inference

**SLEEPM (Sleep Mode)**

* **Input**: 4-bit mode immediate
* **Output**: Clock control signals
* **Function**: Power state management
* **Use Case**: Connectivity power saving

**6. Performance Features**

**Single-Cycle Operations**

* All basic ALU operations
* Custom AI instructions
* Register-register operations

**Multi-Cycle Operations**

* Memory access (load/store)
* Multi-cycle multiplication (in MAC unit)

**Throughput**

* **Target**: 1 instruction/cycle (ideal)
* **Reality**: 0.8-0.9 instructions/cycle (with hazards)

**7. Power Management**

**Clock Gating**

* Pipeline stage enables
* Functional unit enables
* Sleep modes via SLEEPM instruction

**Dynamic Power Saving**

* Inactive custom units powered down
* Memory access minimization
* 16-bit instructions reduce fetch power

**8. Physical Design Considerations**

**Area Optimization**

* 16-bit instruction format
* 16-register file
* Minimal custom unit sizes

**Timing Constraints**

* Critical path: Register file → ALU → Data memory
* Target frequency: 200-500 MHz (low power)
* Pipeline balanced across stages

**9. Verification Features**

**Debug Support**

* Pipeline register visibility
* Control signal monitoring
* Custom unit status outputs

**Testability**

* Scan chain insertion
* BIST (Built-In Self Test) for memories
* Custom unit test modes

**Summary**

This microarchitecture implements a **power-optimized, AI-enhanced pipelined processor** specifically designed for low-cost mobile devices. The design balances performance with the cost and power constraints identified in the domain analysis, while providing hardware acceleration for voice recognition, biometric security, and intelligent connectivity workloads.

**Key Innovations:**

1. Custom AI instructions for domain-specific acceleration
2. Power-aware design with sleep modes
3. Hazard handling for efficient pipelining
4. Cost-optimized 16-bit instruction format
5. Balanced 5-stage pipeline for target workloads

This microarchitecture successfully implements the ISA specification while meeting all design constraints for the target application domain.