**The Processing Unit**

* **Central processing unit**
  + Consists of:
    - Control unit
    - Arithmetic & logic unit
    - Registers
  + In order to execute an instruction, the CPU performs these steps (5 stages):
    - 1. Fetch the next instruction (instruction fetch phase)
      * i.e. IR ← [[PC]]
      * Also PC ← [PC] + 4
    - Perform the operation (instruction execution phase)
      * 2. Decode the instruction to determine operation & fetch operands
      * 3. Execute the operation
      * 4. Read from/write to memory (if applicable)
      * 5. Write back data to register file
* **Components of datapath of RISC processor:**
  + Instruction address generator
    - Contains PC − stores the address of next instruction
    - Increments PC after every instruction
    - Also generates branch & subroutine addresses
  + IR – instruction register stores the current instruction
  + Register file – array of registers
    - Inputs:
      * Address A, address B – specify which 2 registers to read
      * Address C – specify which register to write to
      * C – value of specified dest reg
    - Outputs:
      * A, B – values of specified source regs
    - Address signals must have enough address space for all registers; e.g. 32 regs = 5 bits
  + Processor-memory interface – i.e. caches
    - Used for load/store instructions
    - Inputs:
      * Data (to be stored)
      * Address (to be accessed)
      * Read/write signal
    - Outputs:
      * Data (read)
      * Memory operation done signal
  + ALU – performs arithmetic & logical operations
    - Inputs: operand A, operand B
    - Output: result
* To execute each stage in one cycle, use inter-stage registers
* **Datapath of RISC processor:**
  + Fetch stage
    - Adder – computes next address
      * Inputs: PC, address offset
      * Output: new PC
    - MUX INC selects 4 or branch offset (extracted from IR) → adder offset input
    - MUX PC selects adder output or subroutine address in RA → PC
    - PC → PC-temp → MUX Y for return address
    - MUX MA selects PC → memory address input
    - Memory data output → IR
  + Decode stage (register file)
    - MUX C selects R-type encoding, I-type coding, or link register → address C
    - Output A → RA
    - Output B → RB
  + Execute stage (ALU)
    - RA → operand A
    - MUX B selects RB or immediate value (extracted from IR) → operand B
    - ALU result → RZ
    - If operation = call subroutine, RA → PC
    - If operation = store, transfer RB → RM
  + Memory stage (memory interface)
    - If non-memory operation
      * RZ = computational result
      * MUX Y selects RZ → RY
    - If operation = load
      * RZ = effective address
      * MUX MA selects RZ → memory address input
      * MUX Y selects memory data output → RY
    - If operation = store
      * RZ = effective address
      * MUX MA selects RZ → memory address input
      * RM = data to be stored → memory data input
    - If operation = return from subroutine
      * MUX Y selects PC-temp (return address) → RY
  + Write-back stage (register file)
    - RY → input C of register file
* Control signals
  + Orchestrate the flow of data through the datapath components
  + RF\_write
    - Input to register file
    - = 0 if reading; = 1 if writing
  + C\_select
    - Select lines for MUX C
    - Selects between R-type & I-type instruction encoding & linkage address
  + B\_select
    - Select line to MUX B
    - Selects between RB and immediate value
  + ALU\_op
    - k-bit input to ALU
    - Control code for the operation to be performed (AND, OR, etc.)
  + Condition signals
    - Output from ALU
    - Monitored by control circuitry to ensure branching
  + MEM\_read
    - Input to memory interface
  + MEM\_write
    - Input to memory interface
  + IR\_enable
    - Input to IR
    - Loads new instruction into IR after MFC is asserted
  + MA\_select
    - Select lines for MUX MA
    - Selects between RZ and PC for memory address to send to memory interface
  + Memory function completed (MFC) (stall signal)
    - Output from memory interface
    - If cache hit – MFC is set in the same clock cycle as request
    - If cache miss – MFC is held until completion of request – stalls instruction execution
  + INC\_select
    - Select line for MUX INC
    - Selects between 4 and branch offset
  + PC\_select
    - Select line MUX PC
    - Selects between RA & output from adder if PC\_enable is on
  + PC\_enable
    - Input to PC
* Control signals can be generated using:
  + Hardwired control
    - Instruction decoder decodes opcode & addressing mode in IR → indicates which instruction is being executed
    - Step counter → indicates which stage of execution is being carried out
    - Combinatorial circuit takes these info & external inputs & condition signals to generate control signals
  + Microprogrammed control – i.e. through software