

# Advanced Computer Architecture

Daniel Weaver  
ra18837

# Instruction Set

MOV Rd #imm - Rd <- #imm

MOV Rd Rc - Rd <- Rc

ADD Rd #imm - Rd <- Rd + #imm

ADD Rd Rs Rn - Rd <- Rs + Rn

SUB Rd #imm - Rd <- Rd - #imm

SUB Rd Rs Rn - Rd <- Rs - Rn

MUL Rd Rs Rn - Rd <- Rs \* Rn

DIV Rd Rs Rn - Rd <- Rs / Rn

CMP Rd Rs - -1 if Rd < Rs, 0 if Rd = Rs, +1 if Rd > Rs

LDR Rd [Rb #imm] - Rd <- MEM[Rb + #imm]

LDR Rd [Rb Ro] - Rd <- MEM[Rb + Ro]

STR Rd [Rb #imm] - MEM[Rb + #imm] <- Rd

STR Rd [Rb Ro] - MEM[Rb + Ro] <- Rd

BEQ label - if CMP = 0 goto label

BNE label - if CMP != 0 goto label

BLT label - if CMP = -1 goto label

BGT label - if CMP = +1 goto label

B label - goto label

NOP - No operation

HALT - Stop executing

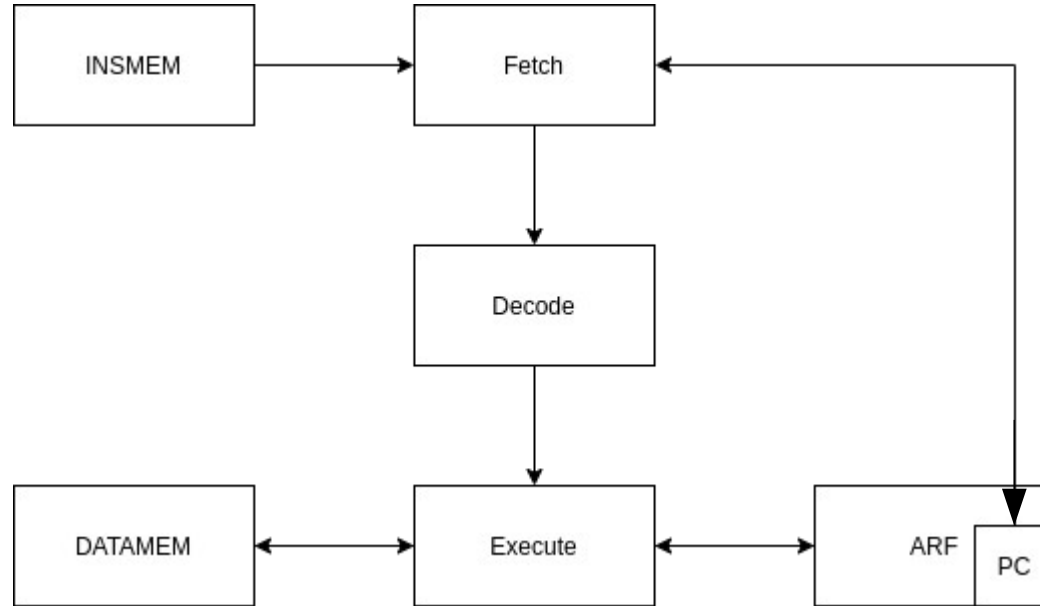
The following are not currently implemented but may be added if any programs need them, call and return instructions may also be implemented storing the state of the processor on the stack. Shifts may also be implemented if necessary.

AND Rd Rs - Rd <- Rd . Rs bitwise

OR Rd Rs - Rd <- Rd + Rs bitwise

NOT Rd - Rd <- NOT Rd bitwise

# Current Processor Diagram



# Features

- In Order
- Non-pipelined
- Scalar

Later on in the project I would like to add (on top of completing stage 3)

- Simple branch prediction e.g. 2 bit dynamic
- Reservation Stations
- Re-order buffer

And if there is time possibly implement SIMD