DEPARTMENT OF ELECTRONIC AND TELECOMMUNICATION

UNIVERSITY OF MORATUWA

EN3030: CIRCUITS AND SYSTEMS DESIGN


# INSTRUCTION SET ARCHITECTURE


| | |
|---|---|
| Mendis N. P. A. | 180398A |
| Kumarasinge H. A. N. H. | 180337M |
| Nagasinghe K.R.Y | 180411K |
| Thalagala B. P. | 180631J |


Date of submission: 15th December 2021

# 1. INSTRUCTION SET

## 1.1. PROGRAM CONTROL

*START/ INITIALIZE*

1. PC ← 0
2. IR ← 0

*FETCH*

1. AR ← PC
2. DR ←M , PC ←PC+1
3. IR ← DR , AR ← PC

*NOP*

IDLE processor

*CLAC*

1. AC ← 0 , Z=1

*ENDOP*

End all operations

### JUMP INSTRUCTIONS

*JUMP*

1. READ
2. AC ← IM(τ)
3. PC ← AC

*JMPZ*

1. READ
2. AC ← IM(τ)
3. PC ← AC
4. PC ←PC+1
5. READ

*JMPNZ*

1. AC ← IM(τ)
2. PC ← AC
3. PC ←PC+1

## 1.2. LOAD AND STORE INSTRUCTIONS

*LDIAC*

1. MEM READ
2. AC ← IM(τ)
3. PC ←PC+1

*LDAC*

1. AC ← AC
2. READ
3. DR ← M(τ)
4. AC ← DR

*STAC*

1. READ AC to bus
2. AR ← AC
3. M ← AC

## 1.3. MOVE INSTRUCTIONS

*MOVR*

1. R ← AC

*MOVACR1*

1. R1 ← AC

*MOVACR2*

1. R2 ← AC

*MOVACR3*

1. R3 ← AC

*MOVACR4*

1. R4 ← AC

*MOVACR5*

1. R5 ← AC

*MOVRAC*

1. AC ← R

*MOVR1AC*

1. AC ← R1

*MOVR2AC*

1. AC ← R2

*MOVR3AC*

1. AC ← R3

*MOVR4AC*

1. AC ← R4

*MOVR5AC*

1. AC ← R5

*MOVAC*

1. PC ← AC
2. AR ← PC

## 1.4. ARITHMETIC AND LOGICAL OPERATIONS

### ALU BASED

*ADD*

1. AC ← AC+R

*SUB*

1. AC ← AC-R

*MULTIPLY*

1. AC ← AC*R

*DIVIDE*

1. AC ← AC/R

*AND*

1. AC ← AC & R

*XOR*

1. AC ← AC XOR R

*OR*

1. AC ← AC|R

*NOT*

1. AC ← $\overline{AC}$

*LSHIFT*

1. AC ← AC<<R

*RSHIFT*

1. AC ← AC>>R

### DEDICATED ADDER BASED

*INCREMENT PC*

1. PC ← PC+1

*INCREMENT AC*

1. AC ← AC+1

*INCREMENT R*

1. PC ← PC+1

*INCREMENT R1*

1. R1 ← R1+1

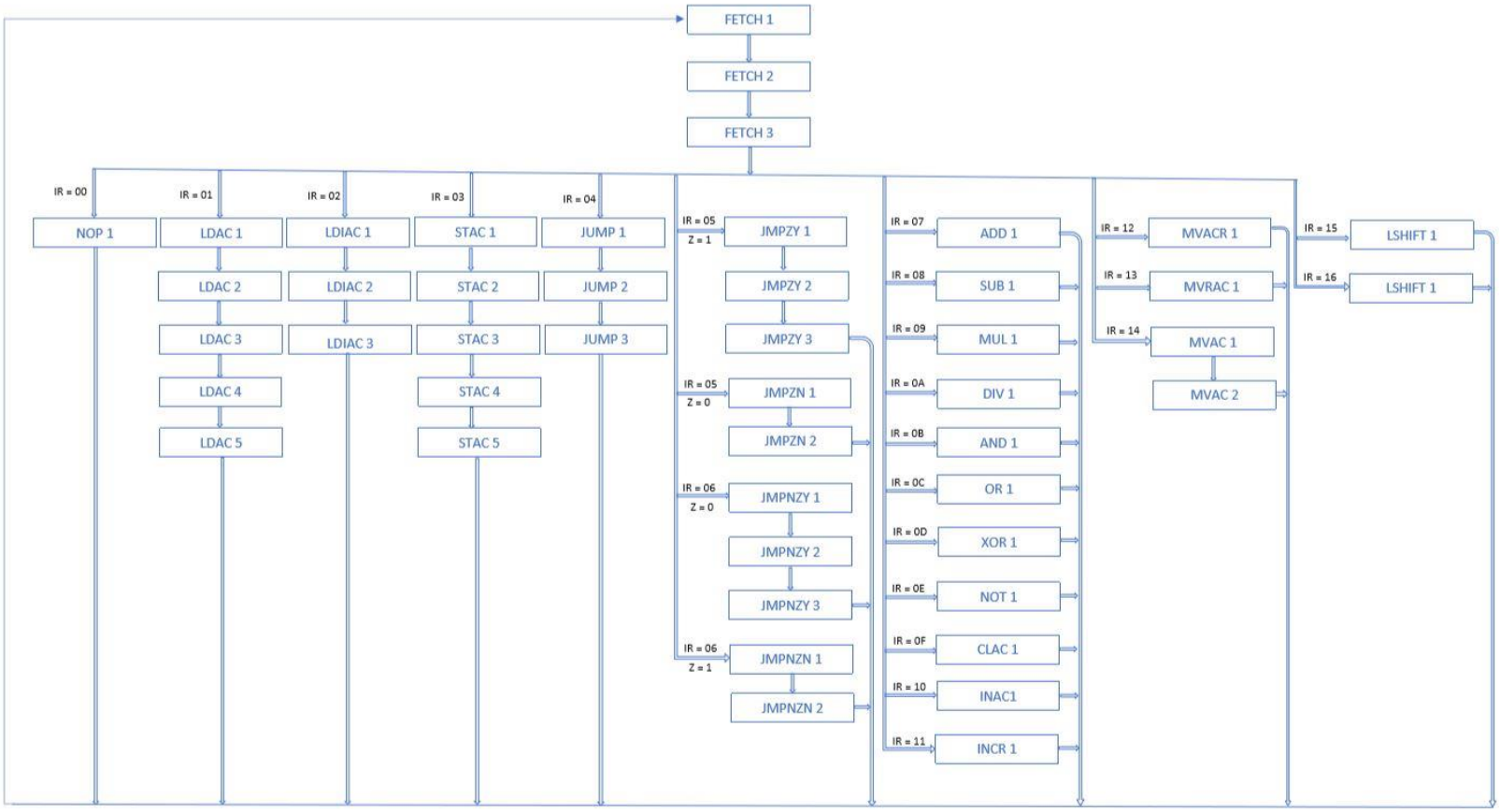
*INCREMENT R2*

1. R2 ← R2+1

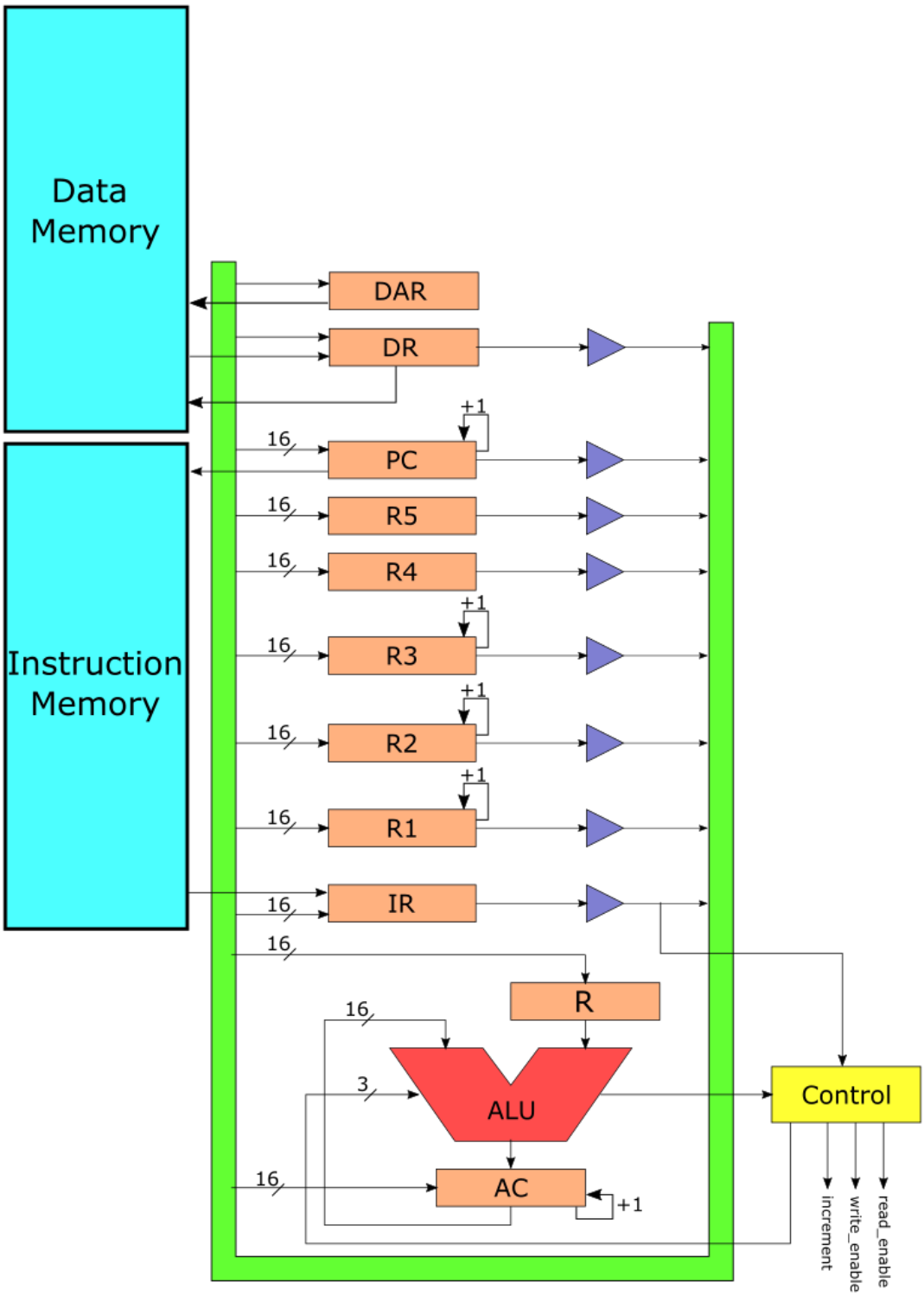
*INCREMENT R3*

1. R3← R3+1

## 2. STATE DIAGRAM



## 3. DATA PATH


Data Memory
Instruction Memory
DAR
DR
PC
R5
R4
R3
R2
R1
IR
R
ALU
AC
Control
+1
16
3
increment
write_enable
read_enable