CMPN301 – Computer Architecture

Project Phase 1 Report

Team number: C4

Team Members:

Omar Mostafa Fouad	1190430
Omar Alaa Eldin	1190377
Hla Hany Mohamed	1190344
Rana Gamal Mostafa	1190449

OPCODES OF EACH INSTRUCTION

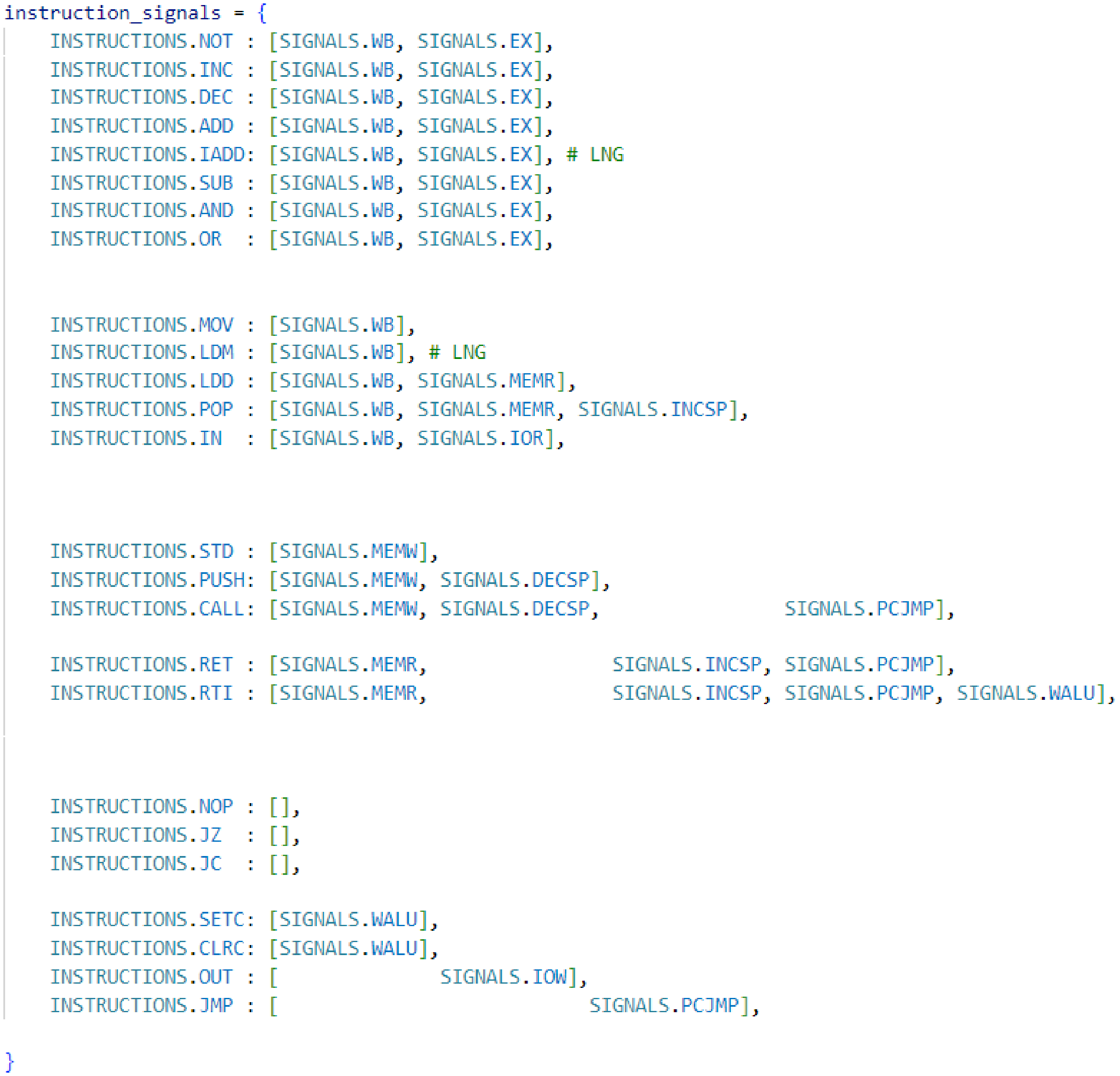
INC	0	0	0	0	0	0	X
ADD	0	0	0	0	0	1	X
IADD	1	0	0	0	0	1	X
SUB	0	0	0	0	1	0	X
DEC	0	0	0	0	1	1	X
AND	0	0	0	1	0	1	X
OR	0	0	0	1	1	0	X
NOT	0	0	0	1	1	1	X
MOV	0	0	1	0	0	0	X
LDM	1	0	1	0	0	1	X
LDD	0	0	1	0	1	0	X
POP	0	0	1	0	1	1	X
IN	0	0	1	1	0	0	X
STD	0	1	0	0	0	0	X
PUSH	0	1	0	0	1	0	X
CALL	0	1	0	0	1	1	X
RET	0	1	0	1	0	1	X
RTI	0	1	0	1	1	1	X
NOP	0	1	1	0	0	0	X
JZ	0	1	1	0	0	1	X
JC	0	1	1	0	1	0	X
SETC	0	1	1	1	0	0	X
CLRC	0	1	1	1	0	1	X
OUT	0	1	1	1	1	0	X
JMP	0	1	1	1	1	1	X

INSTRUCTION BITS DETAILS

15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPCODE							Rdst			Rsrc1			Rsrc2		

15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IMMEDIATE															

CONTROL SIGNALS



```
instruction_signals = {
    INSTRUCTIONS.NOT : [SIGNALS.WB, SIGNALS.EX],
    INSTRUCTIONS.INC : [SIGNALS.WB, SIGNALS.EX],
    INSTRUCTIONS.DEC : [SIGNALS.WB, SIGNALS.EX],
    INSTRUCTIONS.ADD : [SIGNALS.WB, SIGNALS.EX],
    INSTRUCTIONS.IADD: [SIGNALS.WB, SIGNALS.EX], # LNG
    INSTRUCTIONS.SUB : [SIGNALS.WB, SIGNALS.EX],
    INSTRUCTIONS.AND : [SIGNALS.WB, SIGNALS.EX],
    INSTRUCTIONS.OR  : [SIGNALS.WB, SIGNALS.EX],


    INSTRUCTIONS.MOV : [SIGNALS.WB],
    INSTRUCTIONS.LDM : [SIGNALS.WB], # LNG
    INSTRUCTIONS.LDD : [SIGNALS.WB, SIGNALS.MEMR],
    INSTRUCTIONS.POP : [SIGNALS.WB, SIGNALS.MEMR, SIGNALS.INCSP],
    INSTRUCTIONS.IN  : [SIGNALS.WB, SIGNALS.IOR],



    INSTRUCTIONS.STD : [SIGNALS.MEMW],
    INSTRUCTIONS.PUSH: [SIGNALS.MEMW, SIGNALS.DECSP],
    INSTRUCTIONS.CALL: [SIGNALS.MEMW, SIGNALS.DECSP,                SIGNALS.PCJMP],

    INSTRUCTIONS.RET : [SIGNALS.MEMR,                SIGNALS.INCSP, SIGNALS.PCJMP],
    INSTRUCTIONS.RTI : [SIGNALS.MEMR,                SIGNALS.INCSP, SIGNALS.PCJMP, SIGNALS.WALU],



    INSTRUCTIONS.NOP : [],
    INSTRUCTIONS.JZ  : [],
    INSTRUCTIONS.JC  : [],

    INSTRUCTIONS.SETC: [SIGNALS.WALU],
    INSTRUCTIONS.CLRC: [SIGNALS.WALU],
    INSTRUCTIONS.OUT : [              SIGNALS.IOW],
    INSTRUCTIONS.JMP : [                           SIGNALS.PCJMP],

}
```

PIPELINE REGISTERS

IF/ID Buffer: (48 bits)

(16-bit IN) (32-bit Instruction)

ID/EX Buffer: (92 bits)

(16-bit IN) (32-bit instruction) (12-bit signals) (16-bit dataout1) (16-bit dataout2)

EX/MEM1 Buffer: (74 bits)

(16-bit input) (1-bit IOR) (3-bit Rdst) (2-bit stackRW) (1-bit IOW) (1-bit MEMW) (1-bit MEMR) (1-bit WB) (16-bit aluOut) (16-bit registerOut1) (16-bit registerOut2)

MEM1/MEM2 Buffer: (40 bits)

(16-bit input) (1-bit IOR) (1-bit MEMW) (1-bit MEMR) (3bit Rdst) (1-bit IOW) (1-bit WB) (16-bit out)

MEM2/WB Buffer: (38 bits)

(16-bit input) (1-bit IOR) (3-bit Rdst) (1-bit IOW) (1-bit WB) (16-bit out)

PIPELINE HAZARDS

Hazards	Details
Data Hazards	**Forwarding unit at ALU:** Inputs: Rsrc1, Rsrc2 from Decoder, Rdst_EX, Rdst_MEM Outputs: Selector of the ALU operands
Structural Hazards	**Hazard Detection unit:** Inputs: MEMR & MEMW from Data Cache buffer-1 Outputs: Stall the pipeline
Control Hazards	**Static Branch Prediction:** Predict untaken If taken: jump to PC and flush IF/ID & ID/EX buffers
	JMP unconditional Flush IF/ID buffer
	RET/RTI Change PC to Data Memory[SP]