

Project 1

ISA and Controller Micro-Instructions

General Processor Description

- 32 bit data path
- Word length 4 bytes
- Word Addressable only
- 32 registers
- On startup, the stack pointer is initialized to the maximum address **0x2710**

Control Unit ROM Description

- 12 bit address width

OPCODE (B ₁₁ -B ₆) (6-bits)	\$Z (B ₅) (1-bit)	Next Sate (from state register) (B ₀₄ -B ₀₀) (5-bits)
---	----------------------------------	---

- 24 bit data width
 - ROM DATA FORMAT

DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG			FUNC	REGSEL	Next State

- ROM is initialized at address 0

ALU Function Codes

• ADD	000
• A + 1	001
• A – B	010
• MULTIPLY	011
• SHIFT LEFT (LOGICAL)	100
• SHIFT RIGHT (LOGICAL)	101
• AND	110
• OR	111

Register Conventions

This processor uses the same register convention as MIPS.

Number	Name	Purpose
\$0	\$0	Always 0
\$1	\$at	The <i>Assembler Temporary</i> used by the assembler in expanding pseudo-ops
\$2-\$3	\$v0-\$v1	These registers contain the <i>Returned Value</i> of a subroutine.
\$4-\$7	\$a0-\$a3	The <i>Argument</i> registers, these registers contain the first 4 argument values for a subroutine call
\$8-\$15, \$24-\$25	\$t0-\$t9	The <i>Temporary</i> registers.
\$16-\$23	\$s0-\$s7	The <i>Saved</i> Registers.
\$26-\$27	\$k0-\$k1	The <i>Kernel Reserved</i> registers. DO NOT USE
\$28	\$gp	The <i>Globals Pointer</i> used for addressing static global variables.
\$29	\$sp	The <i>Stack Pointer</i>
\$30	\$fp	The <i>Frame Pointer</i>
\$31	\$ra	The <i>Return Address</i> in a subroutine call.

ISA Description

Instruction Types

- R-Type

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₂₁ (5-bits)	B ₂₀ -B ₁₆ (5-bits)	B ₁₅ -B ₁₁ (5-bits)	B ₁₀ -B ₀ (11-bits)
OPCODE	\$R _b	\$R _c	\$R _d	Unused

- I-Type

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₂₁ (5-bits)	B ₂₀ -B ₁₆ (5-bits)	B ₁₅ -B ₀ (16-bits)
OPCODE	\$R _b	\$R _d	Immediate

- J-Type

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₀ (26-bits)
OPCODE	Target Address

- O-Type

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₀ (26-bits)
OPCODE	Unused

Instructions

FETCH (Not callable by the programmer)

- FETCH**

Micro-Instructions

- **FETCH-1** **MAR ← \$PC**
A ← \$PC
 - DrPC
 - LdMAR
 - LdA
- **FETCH-2** **\$PC ← \$PC + 1**
 - DrALU
 - LdPC
 - func 001
- **FETCH-3** **IR ← MEM[MAR]**
 - DrMEM
 - LdIR

FETCH		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		(B ₀₉ -B ₀₇) (3-bits)	(B ₀₆ -B ₀₅) (2-bits)	(B ₀₄ -B ₀₀) (5-bits)	
	curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
FETCH-1	0000 0000 0000	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	000	00	0 0001
FETCH-2	0000 0000 0001	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	001	00	0 0010
FETCH-3	0000 0000 0010	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	000	00	0 0000

curr state	addr	HEX INSTRUCTION
FETCH- 1	0x 000	0x2050001
FETCH- 2	0x 001	0x1080082
FETCH- 3	0x 002	0x0408000

HALT

- **HALT**

- halt;
- Opcode: 00 1110
- O-Type

Instruction Format

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₀ (26-bits)
00 1110	Unused

Micro-Instructions

- **HALT-1**
 - **Blocks Next State from being loaded by the Control Unit**

HALT		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		(B ₀₉ -B ₀₇) (3-bits)	(B ₀₆ -B ₀₅) (2-bits)	(B ₀₄ -B ₀₀) (5-bits)	
		curr state	Addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL
HALT-1	0011 1000 0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	111	00	0 0000

Note: Halt triggers when func is 111 and next state is set to 0 0000.

curr state	addr	HEX INSTRUCTION
HALT- 1	0x 380	0x0000780

Arithmetic (Arithmetic operations do not detect or handle overflows)

- **ADD**

- add, \$R_d, \$R_b, \$R_c;
- $\$R_d \leftarrow \$R_b + \$R_c$
- Opcode: 00 0001
- R-Type

Instruction Format

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₂₁ (5-bits)	B ₂₀ -B ₁₆ (5-bits)	B ₁₅ -B ₁₁ (5-bits)	B ₁₀ -B ₀ (11-bits)
00 0001	\$R _b	\$R _c	\$R _d	xxx xxxx xxxx

Micro-Instructions

- **Add-1** $A \leftarrow \$R_b$
 - DrREG
 - LdA
 - REGSEL 00
- **Add-2** $B \leftarrow \$R_c$
 - DrREG
 - LdB
 - REGSEL 01
- **Add-3** $\$R_d \leftarrow A + B$
 - DrALU
 - WrREG
 - REGSEL 10
 - func 000
- **Add-4** **Reset IR**
 - ResIR

ADD		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
ADD-1	0000 0100 0000	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	000	00	0 0001
ADD-2	0000 0100 0001	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	000	01	0 0010
ADD-3	0000 0100 0010	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	000	10	0 0011
ADD-4	0000 0100 0011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	000	00	0 0000

curr state	addr	HEX INSTRUCTION
ADD-1	0x040	0x0840001
ADD-2	0x041	0x0820022
ADD-3	0x042	0x1001043
ADD-4	0x043	0x0000800

- **ADDi**

- addi, $\$R_d$, $\$R_b$, immediate;
- $\$R_d \leftarrow \$R_b + \text{immediate}$
- Opcode: 00 0010
- I-Type

Instruction Format

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₂₁ (5-bits)	B ₂₀ -B ₁₆ (5-bits)	B ₁₅ -B ₀ (16-bits)
00 0010	$\$R_b$	$\$R_d$	Immediate

Micro-Instructions

- **Addi-1** $A \leftarrow \$R_b$
 - DrREG
 - LdA
 - REGSEL 00
- **Addi-2** $B \leftarrow \text{immediate}$
 - DrIMM
 - LdB
- **Addi-3** $\$R_d \leftarrow A + B$
 - DrALU
 - WrREG
 - REGSEL 01
 - func 000
- **Addi-4** **Reset IR**
 - ResIR

Addi		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
ADDi-1	0000 1000 0000	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	000	00	0 0001
ADDi-2	0000 1000 0001	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	000	00	0 0010
ADDi-3	0000 1000 0010	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	000	01	0 0011
ADDi-4	0000 1000 0011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	000	00	0 0000

curr state	addr	HEX INSTRUCTION
ADDi-1	0x080	0x0840001
ADDi-2	0x081	0x0220022
ADDi-3	0x082	0x1001023
ADDi-4	0x083	0x0000800

- **MULTIPLY**

- mul, \$R_d, \$R_b, \$R_c;
- $\$R_d \leftarrow \$R_b * \$R_c$
- Opcode: 00 0011
- R-Type

Instruction Format

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₂₁ (5-bits)	B ₂₀ -B ₁₆ (5-bits)	B ₁₅ -B ₁₁ (5-bits)	B ₁₀ -B ₀ (11-bits)
00 0011	\$R _b	\$R _c	\$R _d	xxx xxxx xxx

Micro-Instructions

- **MUL-1** $A \leftarrow \$R_b$
 - DrREG
 - LdA
 - REGSEL 00
- **MUL-2** $B \leftarrow \$R_c$
 - DrREG
 - LdB
 - REGSEL 01
- **MUL-3** $\$R_d \leftarrow A * B$
 - DrALU
 - WrREG
 - REGSEL 10
 - func 011
- **MUL-4** **Reset IR**
 - ResIR

MUL		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
MUL-1	0000 1100 0000	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	000	00	0 0001
MUL-2	0000 1100 0001	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	000	01	0 0010
MUL-3	0000 1100 0010	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	011	10	0 0011
MUL-4	0000 1100 0011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	000	00	0 0000

curr state	addr	HEX INSTRUCTION
MUL-1	0x0c0	0x0840001
MUL-2	0x0c1	0x0820022
MUL-3	0x0c2	0x10011c3
MUL-4	0x0c3	0x0000800

- **SHIFT LEFT (Logical)**

- $sll, \$R_d, \$R_b, \text{Shift Amount};$
- $\$R_d \leftarrow \$R_b * 2^{\text{Shift Amount}}$
- Opcode: 00 0100
- I-Type

Instruction Format

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₂₁ (5-bits)	B ₂₀ -B ₁₆ (5-bits)	B ₁₅ -B ₀ (16-bits)
00 0100	$\$R_b$	$\$R_d$	Shift Amount

Micro-Instructions

- **SLL-1** **$A \leftarrow \$R_b$**
 - DrREG
 - LdA
 - REGSEL 00
- **SLL-2** **$B \leftarrow \text{Shift Amount}$**
 - DrIMM
 - LdB
- **SLL-3** **$\$R_d \leftarrow A \text{ shift left by } B$**
 - DrALU
 - WrREG
 - REGSEL 01
 - func 100
- **SLL-4** **Reset IR**
 - ResIR

SLL		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
SLL-1	0001 0000 0000	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	000	00	0 0001
SLL-2	0001 0000 0001	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	000	00	0 0010
SLL-3	0001 0000 0010	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	100	01	0 0011
SLL-4	0001 0000 0011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	000	00	0 0000

curr state	addr	HEX INSTRUCTION
SLL-1	0x100	0x0840001
SLL-2	0x101	0x0220022
SLL-3	0x102	0x1001223
SLL-4	0x103	0x0000800

- **SHIFT RIGHT (Logical)**

- srl, \$R_d, \$R_b, Shift Amount;
- $\$R_d \leftarrow \$R_b * 2^{-\text{Shift Amount}}$
- Opcode: 00 0101
- I-Type

Instruction Format

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₂₁ (5-bits)	B ₂₀ -B ₁₆ (5-bits)	B ₁₅ -B ₀ (16-bits)
00 0101	\$R _b	\$R _d	Shift Amount

Micro-Instructions

- **SRL-1** **A ← \$R_b**
 - DrREG
 - LdA
 - REGSEL 00
- **SRL-2** **B ← Shift Amount**
 - DrIMM
 - LdB
- **SRL-3** **\$R_d ← A shift right by B**
 - DrALU
 - WrREG
 - REGSEL 01
 - func 101
- **SRL-4** **Reset IR**
 - ResIR
 -

SRL		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
SLL-1	0001 0100 0000	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	000	00	0 0001
SLL-2	0001 0100 0001	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	000	00	0 0010
SLL-3	0001 0100 0010	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	101	01	0 0011
SLL-4	0001 0100 0111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	000	00	0 0000

curr state	addr	HEX INSTRUCTION
SRL-1	0x140	0x0840001
SRL-2	0x141	0x0220022
SRL-3	0x142	0x10012a3
SRL-4	0x143	0x0000800

Logic

- **AND**

- and, $\$R_d$, $\$R_b$, $\$R_c$;
- $\$R_d \leftarrow \$R_b \text{ and } \$R_c$
- Opcode: 00 0110
- R-Type

Instruction Format

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₂₁ (5-bits)	B ₂₀ -B ₁₆ (5-bits)	B ₁₅ -B ₁₁ (5-bits)	B ₁₀ -B ₀ (11-bits)
00 0110	$\$R_b$	$\$R_c$	$\$R_d$	xxx xxxx xxx

Micro-Instructions

- **AND-1** **$A \leftarrow R_b$**
 - DrREG
 - LdA
 - REGSEL 00
- **AND-2** **$B \leftarrow R_c$**
 - DrREG
 - LdB
 - REGSEL 01
- **AND-3** **$R_d \leftarrow A \text{ and } B$**
 - DrALU
 - WrREG
 - func 110
 - REGSEL 10
- **AND-4** **Reset IR**
 - ResIR

AND		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
AND-1	0001 1000 0000	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	000	00	0 0001
AND-2	0001 1000 0001	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	000	01	0 0010
AND-3	0001 1000 0010	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	110	10	0 0011
AND-4	0001 1000 0011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	000	00	0 0000

curr state	Addr	HEX INSTRUCTION
AND-1	0x 180	0x0840001
AND-2	0x 181	0x0820022
AND-3	0x 182	0x1001343
AND-4	0x 183	0x0000800

- **OR**

- or, $\$R_d, \$R_b, \$R_c$;
- $\$R_d \leftarrow \$R_b \text{ or } \$R_c$
- Opcode: 00 0111
- R-Type

Instruction Format

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₂₁ (5-bits)	B ₂₀ -B ₁₆ (5-bits)	B ₁₅ -B ₁₁ (5-bits)	B ₁₀ -B ₀ (11-bits)
00 0111	$\$R_b$	$\$R_c$	$\$R_d$	xxx xxxx xxx

Micro-Instructions

- **OR-1** $A \leftarrow R_b$
 - DrREG
 - LdA
 - REGSEL 00
- **OR-2** $B \leftarrow R_c$
 - DrREG
 - LdB
 - REGSEL 01
- **OR-3** $R_d \leftarrow A \text{ or } B$
 - DrALU
 - WrREG
 - func 111
 - REGSEL 10
- **OR-4** **Reset IR**
 - ResIR

OR		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
OR-1	0001 1100 0000	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	000	00	0 0001
OR-2	0001 1100 0001	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	000	01	0 0010
OR-3	0001 1100 0010	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	111	10	0 0011
OR-4	0001 1100 0011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	000	00	0 0000

curr state	Addr	HEX INSTRUCTION
OR-1	0x1c0	0x0840001
OR-2	0x1c1	0x0820022
OR-3	0x1c2	0x10013c3
OR-4	0x1c3	0x0000800

Load/Store

- **Load Word**

- lw, \$R_d, Offset(\$R_b);
- $\$R_d \leftarrow \text{MEM}[\$R_b + \text{Offset}]$
- Opcode: 00 1000
- I-Type

Instruction Format

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₂₁ (5-bits)	B ₂₀ -B ₁₆ (5-bits)	B ₁₅ -B ₀ (16-bits)
00 1000	\$R _b	\$R _d	Offset

Micro-Instructions

- **LW-1** $A \leftarrow R_b$
 - DrREG
 - LdA
 - REGSEL 00
- **LW-2** $B \leftarrow \text{Offset}$
 - DrIMM
 - LdB
- **LW-3** $MAR \leftarrow A + B$
 - DrALU
 - LdMAR
 - func 000
- **LW-4** $R_d \leftarrow \text{MEM}[MAR]$
 - DrMEM
 - WrREG
 - REGSEL 01

- **LW-5**
 - **Reset IR**

LW		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
LW-1	0010 0000 0000	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	000	00	0 0001
LW-2	0010 0000 0001	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	000	00	0 0010
LW-3	0010 0000 0010	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	000	00	0 0011
LW-4	0010 0000 0011	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	000	01	0 0100
LW-5	0010 0000 0100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	000	00	0 0000

curr state	Addr	HEX INSTRUCTION
LW-1	0x200	0x0840001
LW-2	0x201	0x0220002
LW-3	0x202	0x1010003
LW-4	0x203	0x0401024
LW-5	0x204	0x0000800

- **Store Word**

- $sw, \$R_d, \text{Offset}(\$R_b);$
- $\$R_d \leftarrow \text{MEM}[\$R_b + \text{Offset}]$
- Opcode: 00 1001
- I-Type

Instruction Format

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₂₁ (5-bits)	B ₂₀ -B ₁₆ (5-bits)	B ₁₅ -B ₀ (16-bits)
00 1001	$\$R_b$	$\$R_d$	Offset

Micro-Instructions

- **SW-1** **$A \leftarrow R_b$**
 - DrREG
 - LdA
 - REGSEL 00
- **SW-2** **$B \leftarrow \text{Offset}$**
 - DrIMM
 - LdB
- **SW-3** **$MAR \leftarrow A + B$**
 - DrALU
 - LdMAR
 - func 000
- **SW-4** **$\text{MEM}[MAR] \leftarrow R_d$**
 - DrREG
 - WrMEM
 - REGSEL 01
- **SW-5** **Reset IR**
 - ResIR

SW		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
SW-1	0010 0100 0000	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	000	00	0 0001
SW-2	0010 0100 0001	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	000	00	0 0010
SW-3	0010 0100 0010	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	000	00	0 0011
SW-4	0010 0100 0011	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	000	01	0 0100
SW-5	0010 0100 0100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	000	00	0 0000

curr state	Addr	HEX INSTRUCTION
SW-1	0x240	0x0840001
SW-2	0x241	0x0220002
SW-3	0x242	0x1010003
SW-4	0x243	0x0802024
SW-5	0x244	0x0000800

Jumps

- **JUMP**

- j, Target Address;
- $\$PC \leftarrow \text{Target Address}$
- Opcode: 00 1010
- J-Type

Instruction Format

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₀ (26-bits)
00 1010	Target Address (hex)

Micro-Instructions

- **JUMP-1** **PC \leftarrow TARGET**
 - DrTARGET
 - LdPC
- **JUMP-2** **Reset IR**
 - ResIR

JUMP		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
JUMP-1	0010 1000 0000	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	000	00	0 0001
JUMP-2	0010 1000 0001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	000	00	0 0000

curr state	addr	HEX INSTRUCTION
JUMP- 1	0x 280	0x0180001
JUMP- 2	0x 281	0x0000800

- **JUMP REGISTER**

- jr, \$R_b;
- \$PC ← \$R_b
- Opcode: 00 1011
- I-Type

Instruction Format

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₂₁ (5-bits)	B ₂₀ -B ₁₆ (5-bits)	B ₁₅ -B ₀ (16-bits)
00 1011	\$R _b	x xxxx	Unused

Micro-Instructions

- **JR-1** **PC ← \$R_b**
 - DrREG
 - LdPC
 - REGSEL 00
- **JR-2** **Reset IR**
 - ResIR

JR		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
JUMP-1	0010 1100 0000	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	000	00	0 0001
JUMP-2	0010 1100 0001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	000	00	0 0000

curr state	addr	HEX INSTRUCTION
JUMP- 1	0x 2c0	0x0880001
JUMP- 2	0x 2c1	0x0000800

- **JUMP AND LINK**

- jal, Target Address;
- $\$R_a \leftarrow \$PC + 1$
- $\$PC \leftarrow \text{Target Address}$
- Opcode: 00 1100
- J-Type

Instruction Format

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₀ (26-bits)
00 1100	Target Address

***Note:** $\$R_a$ is a register alias for $\$R_{31}$. This register is used for storing return addresses

Micro-Instructions

- **JAL-1** $\$R_a \leftarrow PC$
 - DrPC
 - WrREG
 - REGSEL 11

Note: $\$PC$ is incremented in the fetch instruction

- **JAL-2** $\$PC \leftarrow \text{Target Address}$
 - DrTARGET
 - LdPC
- **JR-3** Reset IR
 - ResIR

JAL		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
JAL-1	0011 0000 0000	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	000	11	0 0001
JAL-2	0011 0000 0001	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	000	00	0 0010
JAL-3	0011 0000 0010	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	000	00	0 0000

curr state	addr	HEX INSTRUCTION
JAL-1	0x300	0x2001061
JAL-2	0x301	0x0180002
JAL-3	0x302	0x0000800

Branch

- **BRANCH ON EQUAL**

- beq, \$R_b, \$R_c, Signed Offset;
- If (\$R_b == \$R_c)
 - $\$PC \leftarrow \$PC + 1 + \text{Signed Offset}$
- Opcode: 00 1101
- I-Type

Instruction Format

B ₃₁ -B ₂₆ (6-bits)	B ₂₅ -B ₂₁ (5-bits)	B ₂₀ -B ₁₆ (5-bits)	B ₁₅ -B ₀ (16-bits)
00 1101	\$R _b	\$R _c	Signed Offset

Micro-Instructions

- **BEQ-1** $A \leftarrow \$R_b$
 - DrREG
 - LdA
 - REGSEL 00
- **BEQ-2** $B \leftarrow \$R_c$
 - DrREG
 - LdB
 - REGSEL 01
- **BEQ-3** $Z \leftarrow A - B$
 - DrALU
 - LdZ
 - func 010

IF (\$R_b == \$R_c)

- **BEQ-4** **A ← PC**
 - DrPC
 - LdA

- **BEQ-5** **B ← IMM**
 - DrIMM
 - LdB

- **BEQ-6** **PC ← A + B**
 - DrALU
 - LdPC
 - func 000

- **BEQ-7** **Reset IR**
 Reset Z
 - ResIR
 - ResZ

BEQ		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
BEQ-1	0011 0100 0000	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	000	00	0 0001
BEQ-2	0011 0100 0001	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	000	01	0 0010
BEQ-3	0011 0100 0010	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	010	00	0 0011
IF (\$R _b == \$R _c) NOTE: Address changed b/c Z == 1																				
BEQ-4	0011 0110 0011	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	000	00	0 0100
BEQ-5	0011 0110 0100	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	000	00	0 0101
BEQ-6	0011 0110 0101	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	000	00	0 0110
BEQ-7	0011 0110 0110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	000	00	0 0000

IF (\$R_b != \$R_c)

BEQ		DRIVE SIGNALS (B ₂₅ -B ₂₀) (6-bits)						LOAD SIGNALS (B ₁₉ -B ₁₄) (6-bits)						WRITE SIGNALS (B ₁₃ -B ₁₂) (2-bits)		RESET SIGNALS (B ₁₁ -B ₁₀) (2-bits)		B ₀₉ -B ₀₇ (3-bits)	B ₀₆ -B ₀₅ (2-bits)	B ₀₄ -B ₀₀ (5-bits)
curr state	addr	PC	ALU	REG	MEM	IMM	TARGET	PC	A	B	MAR	IR	Z	MEM	REG	IR	Z	FUNC	REGSEL	Next State
BEQ-4	0011 0100 0011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	000	00	0 0000

curr state	addr	HEX INSTRUCTION
BEQ-1	0x 340	0x0840001
BEQ-2	0x 341	0x0820022
BEQ-3	0x 342	0x1004103
IF (\$Rb == \$Rc) NOTE: Address changed b/c Z == 1		
BEQ-4	0x 363	0x2040004
BEQ-5	0x 364	0x0220005
BEQ-6	0x 365	0x1080006
BEQ-7	0x 366	0x0000C07

IF (\$Rb != \$Rc)

BEQ-5	0x 343	0x0000C07
-------	-----------	-----------