Robert Pierce
CS 2200
06/01/2016

Project 1

ISA and Controller Micro-Instructions

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

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	$\$R_b$	$\$R_c$	Immediate

- J-Type

B_{31}-B_{26} (6-bits)	B_{25}-B_0 (26-bits)
OPCODE	Target Address

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	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	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	000	00	0 0000

curr state	addr	HEX INSTRUCTION
FETCH-1	0x000	0x2050001
FETCH-2	0x001	0x1080082
FETCH-3	0x002	0x0408000

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	0x 040	0x0840001
ADD-2	0x 041	0x0820022
ADD-3	0x 042	0x1001043
ADD-4	0x 043	0x0000800

- **ADDi**
  - addi $\$R_d$, $\$R_b$, immediate
  - $\$ R_d \leftarrow \$ R_b$ + 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 ← $\$R_b$**
  - DrREG
  - LdA
  - REGSEL 00
- **Addi-2** **B ← immediate**
  - DrIMM
  - LdB
- **Addi-3** **$\$R_d$ ← 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	0x 080	0x0840001
ADDi-2	0x 081	0x0220022
ADDi-3	0x 082	0x1001023
ADDi-4	0x 083	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 xxxx

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	0x 0c0	0x0840001
MUL-2	0x 0c1	0x0820022
MUL-3	0x 0c2	0x10011c3
MUL-4	0x 0c3	0x0000800

- **SHIFT LEFT (Logical)**
  - sll $\$R_d$, $\$R_b$, 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 ← $\$R_b$**
  - DrREG
  - LdA
  - REGSEL 00
- **SLL-2** **B ← Shift Amount**
  - DrIMM
  - LdB
- **SLL-3** **$\$R_d$ ← A 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	0x 100	0x0840001
SLL-2	0x 101	0x0220022
SLL-3	0x 102	0x1001223
SLL-4	0x 103	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** **$A \leftarrow \$R_b$**
  - DrREG
  - LdA
  - REGSEL 00
- **SRL-2** **B ← Shift Amount**
  - DrIMM
  - LdB
- **SRL-3** **$\$R_d \leftarrow$ 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	0x 140	0x0840001
SRL-2	0x 141	0x0220022
SRL-3	0x 142	0x10012a3
SRL-4	0x 143	0x0000800

Logic

- **AND**
  - and $\$R_d$, $\$R_b$, $\$R_c$
  - $\$ R_d \leftarrow \$ R_b$ 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 xxxx

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 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 $\$R_d$, $\$R_b$, $\$R_c$
  - $\$ R_d \leftarrow \$ R_b$ 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 xxxx

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 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	0x 1c0	0x0840001
OR-2	0x 1c1	0x0820022
OR-3	0x 1c2	0x10013c3
OR-4	0x 1c3	0x0000800

Load/Store

- **Load Word**
  - lw R_d, Offset(R_b)
  - $ R_d ← MEM[R_b + 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 ← R_b**
  - DrREG
  - LdA
  - REGSEL 00
- **LW-2** **B ← Offset**
  - DrIMM
  - LdB
- **LW-3** **MAR ← A + B**
  - DrALU
  - LdMAR
  - func 000
- **LW-4** **R_d ← MEM[MAR]**
  - DrMEM
  - WrREG
  - REGSEL 01

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

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	0x 200	0x0840001
LW-2	0x 201	0x0220002
LW-3	0x 202	0x1010003
LW-4	0x 203	0x0401024
LW-5	0x 204	0x0000800

- **Store Word**
  - lw R_d, Offset(R_b)
  - $ R_d ← MEM[R_b + 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

- **SW-1** **A ← R_b**
  - DrREG
  - LdA
  - REGSEL 00
- **SW-2** **B ← Offset**
  - DrIMM
  - LdB
- **SW-3** **MAR ← A + B**
  - DrALU
  - LdMAR
  - func 000
- **SW-4** **MEM[MAR] ← 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	0x 240	0x0840001
SW-2	0x 241	0x0220002
SW-3	0x 242	0x1010003
SW-4	0x 243	0x0802024
SW-5	0x 244	0x0000800

Jumps

- **JUMP**
  - j Target Address
  - $PC ← 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

Micro-Instructions

- **JUMP-1** **PC ← 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	0x280	0x0180001
JUMP-2	0x281	0x0000800

- **JUMP REGISTER**
  - jr $\$R_b$
  - $\$PC \leftarrow \$R_b$
  - 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 ← 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 ← PC**
  - DrPC
  - WrREG
  - REGSEL 11

  Note: $PC is incremented in the fetch instruction

- **JAL-2** **$PC ← 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	0x 300	0x2001061
JAL-2	0x 301	0x0180002
JAL-3	0x 302	0x0000800

Branch

- **BRANCH ON EQUAL**
  - beq $\$R_b$, $\$R_c$, Signed Offset
  - If ($\$R_b$ == $\$R_c$)
    - $PC ← $PC + 1 + 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 ← $\$R_b$**
  - DrREG
  - LdA
  - REGSEL 00
- **BEQ-2** **B ← $\$R_c$**
  - DrREG
  - LdB
  - REGSEL 01
- **BEQ-3** **Z ← 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		**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 ($Rb == $Rc) 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 (R_b$!= R_c$)

BEQ-5	0x 343	0x0000C07