Chapter 1

Cơ sở lý thuyết về CPU RICSV 32

1.1 Giới thiệu tổng quát về CPU RICSV 32

CPU RICSV 32 có tổng cộng 32 lệnh hợp ngữ, mỗi lệnh có độ dài 32 bits và 7 bits [6:0] (opcode) để xác định loại lệnh.

Tập lệnh của RICSV 32 còn được gọi là tập lệnh kiểu load-store, nghĩa là data trong bộ nhớ muốn được thực thi thì trước hết phải được lấy ra bỏ vào băng thanh ghi rồi mới được tính toán. Sau khi tính toán, data sẽ được lưu lại vào memory.

Các thanh ghi trong băng thanh ghi (Register Bank) có độ dài 32 bits và có 32 thanh ghi (từ $x_0 - x_{31}$), cần có 5 bits để xác định địa chỉ của các thanh ghi trong băng thanh ghi. Trong đó, chức năng của 32 thanh ghi được cho như bảng bên dưới.

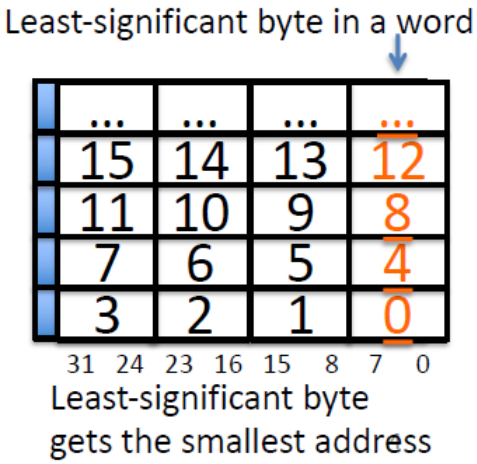
Register	ABI name	Description	Saver
x0	zero	Hard-wired zero	_ _ _
x1	ra	Return address	Caller
x2	sp	Stack pointer	Caller
x3	gp	Global pointer	_ _ _
x4	tp	Thread pointer	_ _ _
x5	t0	Temporary/alternate link register	Caller
x6-7	t1-2	Temporaries	Caller
x8	s0/fp	Saved register/frame pointer	Caller
x9	s1	Saved register	Caller
x10-11	a0-1	Function arguments/return values	Caller
x12-17	a2-7	Function arguments	Caller
x18-27	s2-11	Saved registers	Caller
x28-31	t3-6	Temporaries	Caller

Lưu ý: Thanh ghi x_0 luôn có giá trị bằng 0x000000000 và không thay đổi giá trị này.

Dữ liệu trong cả bộ nhớ dữ liệu (DMEM) và bộ nhớ chương trình (IMEM) đều có độ dài 32bit và được sắp xếp theo kiểu **little edian**.

DMEM được định địa chỉ theo từng byte (= 8 bits) chứ không theo word (= 32 bits). Nếu định địa chỉ theo word thì lấy địa chỉ của byte có trọng số thấp nhất.

Điều này được trình bày như ở hình dưới.



1.2 Nhóm lệnh R-Format

Nhóm lệnh này bao gồm các lệnh có cấu trúc như ở hình sau:

0000000	rs2	rs1	000	rd	0110011	ADD
0100000	rs2	rs1	000	rd	0110011	SUB
0000000	rs2	rs1	001	rd	0110011	SLL
0000000	rs2	rs1	010	rd	0110011	SLT
0000000	rs2	rs1	011	rd	0110011	SLTU
0000000	rs2	rs1	100	rd	0110011	XOR
0000000	rs2	rs1	101	rd	0110011	SRL
0100000	rs2	rs1	101	rd	0110011	SRA
0000000	rs2	rs1	110	rd	0110011	OR
0000000	rs2	rs1	111	rd	0110011	AND

Nhóm lệnh này có opcode là [6:0] = 0110011

Nhóm lệnh này thực hiện lấy hai giá trị lưu ở thanh ghi **rs1** và **rs2** thực hiện đưa vào khối ALU để tính toán, sau đó lưu kết quả vào thanh ghi **rd**.

1.3 Nhóm lệnh I (tính toán)

Nhóm lệnh này có opcode là [6:0] = 0010011

Nhóm lệnh này (trừ 3 lệnh SRAI, SRLI, SLLI) thực hiện lấy giá trị lưu ở thanh ghi **rs1** và giá trị lưu ở **imm[11:0]** (được mở rộng dấu), thực hiện đưa vào khối ALU để tính toán. Kết quả được lưu vào thanh ghi **rd**.

imm[11:0]		rs1	000	rd	0010011	addi
imm[11:0]		rs1	010	rd	0010011	slti
imm[11:0]		rs1	011	rd	0010011	sltiu
imm[11:0]		rs1	100	rd	0010011	xori
imm[11:0]		rs1	110	rd	0010011	ori
imm[11:0]		rs1	111	rd	0010011	andi
0000000	shamt	rs1	001	rd	0010011	slli
0000000	shamt	rs1	101	rd	0010011	srli
0100000	shamt	rs1	101	rd	0010011	srai

One of the higher-order immediate bits is used to distinguish "shift right logical" (SRLI) from "shift right arithmetic" (SRAI)

"Shift-by-immediate" instructions only use lower 5 bits of the immediate value for shift amount (can only shift by 0-31 bit positions)

1.4 Nhóm lệnh L (Load data)

imm[11:0]	rs1	000	rd	0000011	lb
imm[11:0]	rs1	010	rd	0000011	lh
imm[11:0]	rs1	011	rd	0000011	lw
imm[11:0]	rs1	100	rd	0000011	lbu
imm[11:0]	rs1	110	rd	0000011	lhu

funct3 field encodes size and 'signedness' of load data

Nhóm lệnh này có opcode là [6:0] = 0000011

Nhóm lệnh này thực hiện lấy hai giá trị lưu ở thanh ghi **rs1** và giá trị lưu ở **imm[11:0]** (được mở rộng dấu) để tính tổng **rs1 + ext(imm[11:0])**. Sau đó lấy giá trị lưu trong DMEM tại địa chỉ **rs1 + ext(imm[11:0])**, lưu vào thanh ghi **rd**.

1.5 Nhóm lệnh S (Store data)

Imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	sb
Imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	sh
Imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	sw

width

Nhóm lệnh này có opcode là [6:0] = 0100011

Nhóm lệnh này thực hiện lấy hai giá trị lưu ở thanh ghi **rs1** và giá trị lưu ở **imm[11:5]** và **imm[4:0]** (ghép lại và mở rộng dấu) để tính tổng **rs1 + ext(imm[11:5])imm[4:0]**. Sau đó lấy giá trị lưu trong thanh ghi **rs2** lưu vào DMEM tại địa chỉ **rs1 + ext(imm[11:5]imm[4:0])**.

1.6 Nhóm lệnh B (Rẽ nhánh)

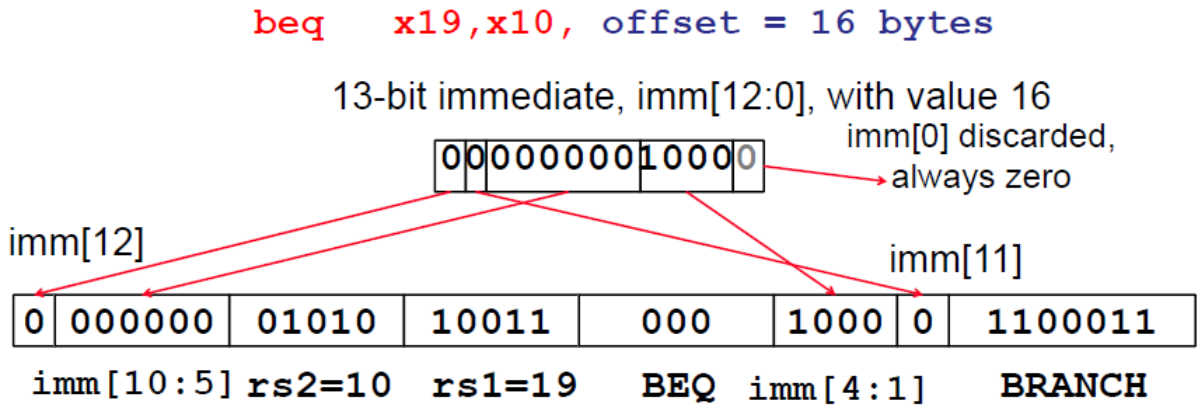
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
imm[12|10:5]	rs2	rs1	001	imm[4:1|11]	1100011	BNE
imm[12|10:5]	rs2	rs1	100	imm[4:1|11]	1100011	BLT
imm[12|10:5]	rs2	rs1	101	imm[4:1|11]	1100011	BGE
imm[12|10:5]	rs2	rs1	110	imm[4:1|11]	1100011	BLTU
imm[12|10:5]	rs2	rs1	111	imm[4:1|11]	1100011	BGEU

Nhóm lệnh này có opcode là [6:0] = 1100011

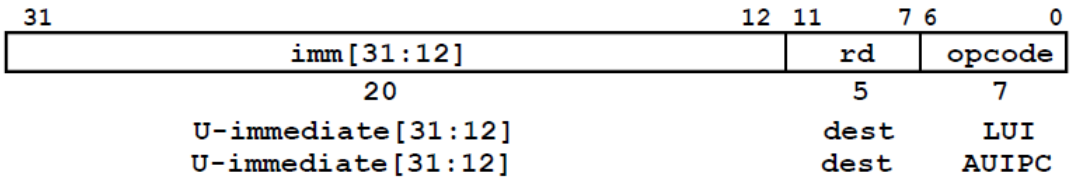
Nhóm lệnh này sẽ thực hiện chuyển giá trị của thanh ghi PC thành giá trị được lưu trong các phần **imm** giá trị lưu trong **rs1** và **rs2** thỏa điều kiện câu lệnh (bằng, không bằng, lớn hơn hoặc bằng,...).

Khi lấy giá trị lưu ở phần **imm** ta phải ghép lại cho đúng thứ tự và mở rộng dấu, bit LSB luôn luôn bằng 0.

Lấy ví dụ như ở hình dưới:



1.7 Nhóm lệnh U



Với lệnh LUI

- Opcode = 0110111
- Lệnh này load giá trị **imm[31:12]00000000000** vào thanh ghi **rd**.

Với lệnh AUIPC

- Opcode = 0010111
- Lệnh này load giá trị ở **PC** vào thanh ghi rd.

1.8 Nhóm lệnh J (nhảy không điều kiện)