ĐẠI HỌC BÁCH KHOA TP.HCM
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

* * * * * * * * * *

Đồ án thiết kế luận lý

HỌC KÌ HK251

Version 0.06

- Ngày 26 tháng 9 năm 2025 -

Changelog

Version 0.06 - 2025/09/26

- Added MIPS processing flow.
- Formatting.

Version 0.05 - 2025/09/26

- Defined bit extraction operation.
- Renamed opcode 0010.
- Fixed mistake in the specification for the *jump* instruction.
- Added instruction for extra work.

Version 0.04 - 2025/09/18

- Added and gave a brief explanation to all required opcodes for now.

Version 0.03 - 2025/09/17

- Added all required opcodes. MTSR and MFSR to be explained in a later version.

Version 0.02 - 2025/09/17

- Added more opcodes.

Version 0.01 - 2025/09/16

- Created document.

GIỚI THIỆU

1. MỤC TIÊU

- Rèn luyện, nâng cao khả năng sử dụng ngôn ngữ đặc tả phần cứng Verilog HDL để thiết kế mạch logic số.
- Rèn luyện kỹ năng phân tích, thiết kế mạch số theo hướng tiếp cận mô hình phân cấp: phân chia chức năng, module, v.v. Nâng cao tính sáng tạo trong thiết kế các sản phẩm ứng dụng hệ thống số.
- Rèn luyện kỹ năng nghiên cứu, tự học, có thêm hiểu biết về các lĩnh vực liên quan: các kỹ thuật trong các mạch điện tử số thông dụng, các lĩnh vực ứng dụng hệ thống số,...

2. YÊU CẦU

- Thực hiện theo nhóm đã được phân công
- Báo cáo tiến độ giữa kỳ trong tuần 45, hình thức online. Trình bày ngắn (khoảng 5 phút) về các nội dung sau đây:
  - Ý tưởng: Phân tích các ý tưởng mà nhóm dự định sẽ thực hiện để đáp ứng yêu cầu đề tài.
  - Đặc tả hệ thống: Xác định các yêu cầu cụ thể của hệ thống.
  - Thiết kế: Sơ đồ khối, chức năng của các khối theo ý tưởng của nhóm.
  - Khó khăn, hướng giải quyết.
- Báo cáo cuối kỳ vào tuần 52. Yêu cầu sử dụng LaTeX để viết báo cáo, và phải nộp file trên LMS cho GVHD ít nhất 2 ngày trước buổi demo và thuyết trình. Bài báo cáo cần có các nội dung sau đây:
  - **Phần 1**: Giới thiệu. Bao gồm các nội dung về giới thiệu đề tài, lý thuyết sơ lược về đề tài, ứng dụng thực tiễn, v.v.
  - **Phần 2**: Thiết kế. Bao gồm sơ đồ khối, các khối chức năng, trình bày chức năng của các khối.
  - **Phần 3**: Hiện thực. Bao gồm cách thức hiện thực thiết kế (Sơ đồ khối, cấu trúc mạch, mô hình thiết kế, v.v.). Có thể vẽ các flow-chart thể hiện luồng xử lý của hệ thống.
  - **Phần 4**: Kiểm thử và đánh giá. Bao gồm kết quả mô phỏng các trường hợp kiểm thử.

Đánh giá thông số hiệu năng về tài nguyên, thời gian, năng lượng, ...

– **Phần 5**: Kết luận. Bao gồm kết luận của nhóm, hướng phát triển trong tương lai, những khó khăn gặp phải và bảng phân chia công việc.

3. TIÊU CHÍ ĐÁNH GIÁ

Tiêu chí đánh giá cụ thể sẽ được cập nhật lại sau.

4. TÀI LIỆU THAM KHẢO

- Slide bài giảng.
- Bài tập thực hành, các tài liệu hướng dẫn.
- Internet

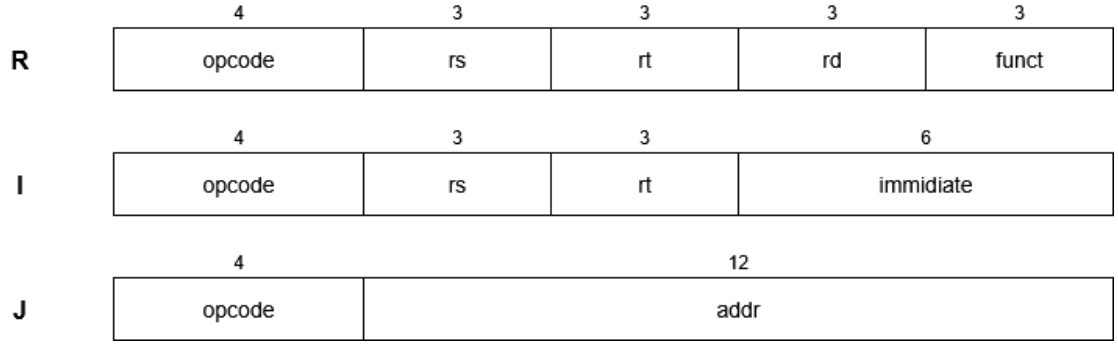
ĐỀ TÀI

Thiết kế RISC CPU dựa trên MIPS

1. Giới thiệu

RISC (Reduced Instructions Set Computer) là một phương pháp thiết kế bộ xử lý hiện nay. Trong đề tài này, chúng ta sẽ thiết kế một RISC CPU 16-bit dựa trên kiến trúc MIPS với 4-bit opcode, 8 thanh ghi đa dụng ($0 - $7) và 6 thanh ghi đặc thù ($ZERO, $PC, $RA, $AT, $HI, $LO). Không gian địa chỉ là 2^{16}.

Tương tự như kiến trúc MIPS, sẽ có 3 kiểu câu lệnh là R, I và J.



Bộ xử lý hoạt động dựa trên tín hiệu *clock* và *reset*. Chương trình sẽ dừng lại khi có câu lệnh HALT.

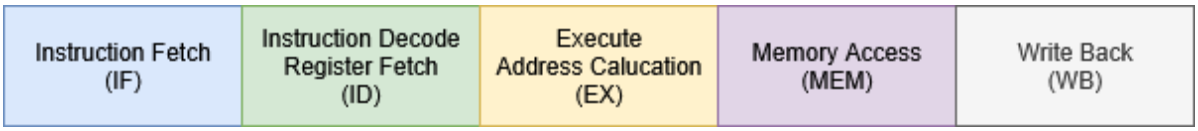
2. Yêu cầu

Yêu cầu cụ thể sẽ được cập nhật lại sau.

Hãy sử dụng các kiến thức về Thiết kế luận lý với HDL, Kiến trúc máy tính và các môn học liên quan để thiết kế CPU theo đặc tả trong file hướng dẫn này trên kit phát triển FPGA Arty-Z7 hoặc tương đương.

3. Chu kì thực thi lệnh

Mỗi câu lệnh trong MIPS được thực thi thông qua tối đa 5 giai đoạn (stages) trong hình dưới:



Mỗi stage được thực hiện trong 1 chu kì của clock, do đó nếu không có pipelining thì mỗi câu lệnh sẽ mất tối đa 5 chu kì clock để hoàn thành.

Miêu tả cho từng stage như sau:

- **IF:** Câu lệnh kế tiếp được lấy từ địa chỉ [$\$PC$] của bộ nhớ.
- **ID:** Decode câu lệnh lấy được trong giai đoạn **IF**; Lấy giá trị của những thanh ghi được yêu cầu; Nạp giá trị kế tiếp cho $\$PC$.
- **EX:** Thực hiện câu lệnh (cụ thể là ALU).
- **MEM:** Đọc hoặc ghi bộ nhớ tùy theo câu lệnh hiện tại.
- **WB:** Ghi kết quả trở lại thanh ghi đích.

Do tập lệnh trong đề tài này phỏng theo kiến trúc tập lệnh MIPS, SV nên hiện thực thiết kế với 5 giai đoạn trên.

4. Đặc tả về tập lệnh

- $\leftarrow$ là kí hiệu cho phép gán (assignment).
- [$\$x$] có nghĩa là "nội dung của thanh ghi $x".
- || là kí hiệu cho phép nối (concatenate).
- $x_{a,b}$ có nghĩa là trích chuỗi bit từ bit có trọng số 2^a đến bit có trọng số 2^b của x. Ví dụ (dấu _ chỉ mang ý nghĩa cách khoảng):

$$(0110\_0101)_{5,1} = 10\_010$$

- Phép so sánh sẽ cho kết quả 1 nếu đúng và 0 nếu ngược lại.
- $SE(x)$ là Sign Extension của x. Nếu x âm thì các bit 1 sẽ được thêm vào bên trái, ngược lại thì là các bit 0.

4.1. Tập lệnh

opcode	Tên	Kiểu	Đặc tả
0000	ALU0	R	Xem Nhóm lệnh ALU0
0001	ALU1	R	Xem Nhóm lệnh ALU1
0010	ALU2	R	Xem Nhóm lệnh Shift (ALU2)
0011	ADDI	I	Xem Nhóm lệnh Immidiate
0100	SLTI	I	
0101	BNEQ	I	Xem Nhóm lệnh Branch
0110	BGTZ	I	
0111	JUMP	J	Xem Lệnh Jump
1000	LH	I	Xem Nhóm lệnh Memory
1001	SH	I	
1010	MFSR	R	Xem Thanh ghi đặc biệt
1011	MTSR	R	
1100	Không xác định		
1101			
1110			
1111	HLT	J	Dừng chương trình

4.2. Nhóm lệnh ALU0

Cho cả 8 funct này, tất cả các giá trị đều được xem là unsigned.

funct	Mã Assembly	Đặc tả
000	addu $rd, $rs, $rt	$\$rd \leftarrow [\$rs] + [\$rt]$
001	subu $rd, $rs, $rt	$\$rd \leftarrow [\$rs] - [\$rt]$
010	multu $rs, $rt	$\$HI \,||\, \$LO \leftarrow [\$rs] * [\$rt]$
011	divu $rs, $rt	$\$LO \leftarrow [\$rs]/[\$rt]$; $\$HI \leftarrow [\$rs]\%[\$rt]$
100	and $rd, $rs, $rt	$\$rd \leftarrow [\$rs]\&[\$rt]$
101	or $rd, $rs, $rt	$\$rd \leftarrow [\$rs] \,|\, [\$rt]$
110	nor $rd, $rs, $rt	$\$rd \leftarrow \tilde{}([\$rs] \,|\, [\$rt])$
111	xor $rd, $rs, $rt	$\$rd \leftarrow [\$rs] \oplus [\$rt]$

4.3. Nhóm lệnh ALU1

Ngoại trừ *sltu* và *jr*, 6 funct còn lại xem các giá trị của thanh ghi là signed theo hệ bù 2.

funct	Mã Assembly	Đặc tả
000	add $rd, $rs, $rt	$\$rd \leftarrow [\$rs] + [\$rt]$
001	sub $rd, $rs, $rt	$\$rd \leftarrow [\$rs] - [\$rt]$
010	mult $rs, $rt	$\$HI \,||\, \$LO \leftarrow [\$rs] * [\$rt]$
011	div $rs, $rt	$\$LO \leftarrow [\$rs]/[\$rt]$; $\$HI \leftarrow [\$rs]\%[\$rt]$
100	slt $rd, $rs, $rt	$\$rd \leftarrow ([\$rs] < [\$rt])$
101	seq $rd, $rs, $rt	$\$rd \leftarrow ([\$rs] == [\$rt])$
110	sltu $rd, $rs, $rt	$\$rd \leftarrow ([\$rs] < [\$rt])$
111	jr $rs	$\$PC \leftarrow [\$rs]$

4.4. Nhóm lệnh Shift (ALU2)

Khác với hàm *shift*, trong hàm *rotate* những bit bị "rơi" ra ngoài ở phía đầu sẽ không mất đi mà sẽ xuất hiện lại ở phía đuôi bên kia. $ROTR$ là *rotate* về phía phải, $ROTL$ là *rotate* về phía trái.

Đọc thêm về hàm *rotate* tại Wikipedia.

funct	Mã Assembly	Tên	Đặc tả
000	shr $rd, $rs, $rt	Shift right	$\$rd \leftarrow [\$rt] >> [\$rs]_{3,0}$
001	shl $rd, $rs, $rt	Shift left	$\$rd \leftarrow [\$rt] << [\$rs]_{3,0}$
010	ror $rd, $rs, $rt	Rotate right	$\$rd \leftarrow ROTR([\$rt], [\$rs]_{3,0})$
011	rol $rd, $rs, $rt	Rotate left	$\$rd \leftarrow ROTL([\$rt], [\$rs]_{3,0})$
100	Không xác định		
101			
110			
111			

4.5. Nhóm lệnh Immidiate

Cho cả hai lệnh này, immidiate đều là signed.

opcode	Mã Assembly	Đặc tả
0011	addi $rt, $rs, i	$\$rt \leftarrow [\$rs] + SE(i)$
0100	slti $rt, $rs	$\$rt \leftarrow ([\$rs] < SE(i))$

4.6. Nhóm lệnh Branch

i ở đây là một số nguyên có dấu.

opcode	Mã Assembly	Tên	Đặc tả
0101	bneq $rs, $rt, i	Branch if not equal	$if\ ([\$rs] \neq [\$rt]) : \$PC \leftarrow [\$PC] + i * 2$
0110	bgtz $rs, i	Branch if greater than 0	$if\ ([\$rs] > 0) : \$PC \leftarrow [\$PC] + i * 2$

4.7. Lệnh Jump

opcode	Mã Assembly	Đặc tả
0111	j addr	$\$PC \leftarrow [\$PC]_{15,13} \,||\, (addr << 1)$

4.8. Nhóm lệnh Memory

Lệnh *lh* load 2 byte từ bộ nhớ đến thanh ghi $rt, còn lệnh *sh* store 2 byte từ thanh ghi $rt vào bộ nhớ.

Không như MIPS, tập lệnh không hỗ trợ đọc hay ghi đơn vị phân chia nhỏ hơn kích thước của câu lệnh, do đó *lh* và *sh* sẽ không xét tới bit có trọng số nhỏ nhất của [$rs].

Công thức để tính địa chỉ bắt đầu *start* dựa trên $rs và giá trị immediate i là:

$$start = ([\$rs]_{15,1} + i) << 1$$

opcode	Mã Assembly	Tên	Đặc tả
1000	lh $rt, i($rs)	Load half-word	$\$rt \leftarrow MEM[start : start + 1]$
1001	sh $rt, i($rs)	Store half-word	$MEM[start : start + 1] \leftarrow \$rt$

4.9. Thanh ghi đặc biệt

Có 6 thanh ghi đặc thù như sau:

Tên	Đặc tả
$\$ZERO$	Giá trị của thanh ghi này luôn luôn là 0x0000.
$\$PC$	*Program Counter*. Giá trị của thanh ghi này là địa chỉ của câu lệnh kế tiếp. Do kiến trúc này là 16-bit, cho nên mỗi bước nhảy của $PC ở đây là 2.
$\$RA$	*Return Address*. Lưu địa chỉ câu lệnh kế tiếp sau khi một hàm thực hiện xong. Do tập lệnh trong file đặc tả này không có *jal*, ta có thể tưởng tượng rằng compiler cho CPU này tự thêm instruction để ghi thanh ghi này khi gọi hàm và đọc thanh ghi này khi trái lại.
$\$AT$	*Assembler Temporary*. Biến tạm dành riêng cho Assembler.
$\$HI$	Dùng cho một số câu lệnh.
$\$LO$	

Nhóm lệnh MFSR

MFSR là viết tắt cho "Move From Special Register".

funct	Mã Assembly	Đặc tả
000	mfz $rd	$\$rd \leftarrow [\$ZERO]$
001	mfpc $rd	$\$rd \leftarrow [\$PC]$
010	mfra $rd	$\$rd \leftarrow [\$RA]$
011	mfat $rd	$\$rd \leftarrow [\$AT]$
100	mfhi $rd	$\$rd \leftarrow [\$HI]$
101	mflo $rd	$\$rd \leftarrow [\$LO]$
110	Không xác định	
111		

Nhóm lệnh MTSR

MFSR là viết tắt cho "Move To Special Register".

funct	Mã Assembly	Đặc tả
000	Không xác định	
001		
010	mtra $rt	$[\$RA] \leftarrow \$rt$
011	mtat $rt	$[\$AT] \leftarrow \$rt$
100	mthi $rt	$[\$HI] \leftarrow \$rt$
101	mtlo $rt	$[\$LO] \leftarrow \$rt$
110	Không xác định	
111		

5. Nội dung làm thêm gợi ý

- Tăng thêm câu lệnh và tính năng vào những vị trí *"Không xác định"* trong đặc tả tập lệnh. Một số ví dụ như các tính toán floating-point (half-precision format - FP16), tính toán số nguyên 32-bit, v.v.
- Vận dụng kiến thức môn học Kiến trúc máy tính để cải thiện thiết kế như thực hiện pipelining, xử lý Hazard, tạo memory cache, v.v.
- Sinh viên có thể đề xuất các nội dung khác.

Việc làm thêm nội dung là không bắt buộc. Tuy nhiên SV được khuyến khích tự tìm tòi và làm thêm nội dung để đồ án được đánh giá cao hơn.

Các nội dung làm thêm nên được trình bày rõ ràng trong report và trong thuyết trình (nếu đủ thời gian) về mặt ý tưởng, thiết kế, hiện thực và kiểm thử.