

CÂU HỎI TRẮC NGHIỆM (9 điểm, 0.3 điểm/câu), SV chọn 1 đáp án đúng

Câu 1 Phép chia 2 số 4bit thì cần bao nhiêu lần lặp (không tính bước khởi tạo)? (G1)

| | | | |
|------|------|------|------|
| A. 5 | B. 6 | C. 7 | D. 8 |
|------|------|------|------|

Câu 2 Thực hiện phép chia $27_{(10)} : 7_{(10)}$ với phần cứng 3 thanh ghi và dùng 6bit để biểu diễn. Hỏi giá trị của thanh ghi thương (quotient) sau khi kết thúc lần lặp thứ 2 là bao nhiêu? (G1)

| | | | |
|-----------------|------------|-----------------|-----------|
| A. 000110110000 | B. 0000000 | C. 000000011011 | D. 000001 |
|-----------------|------------|-----------------|-----------|

Câu 3 Chọn phát biểu SAI trong các phát biểu sau đây? (G1)

| | |
|----|--|
| A. | Dấu của thương sẽ là âm nếu dấu của số chia và số bị chia trái ngược nhau |
| B. | Dấu của số dư được xác định qua công thức: Số dư = Số bị chia – (Thương × Số chia) |
| C. | Lệnh thực hiện phép chia không dấu trong MIPS là div |
| D. | Phần cứng thực hiện phép chia hai số 16 bit phải có ALU xử lý các phép toán 32 bit |

Câu 4 MIPS dùng thanh ghi nào để chứa thương số của phép chia? (G1) [S d trong Hi](#)

| | | | |
|-------|-------|----------------|---------|
| A. Hi | B. Lo | C. Cả Hi và Lo | D. \$ra |
|-------|-------|----------------|---------|

Câu 5 Hãy cho biết đường nào trong các đường sau là critical path của lệnh ‘add’ với datapath như trong hình 1? (G1)

| | |
|----|---------------------------------------|
| A. | I-Mem, Mux, Regs, Mux, ALU, Mux, Regs |
| B. | I-Mem, Mux, Regs, ALU, Mux, Regs |
| C. | I-Mem, Mux, Regs, ALU, Mux, D-Mem |
| D. | I-Mem, Regs, Mux, ALU, Mux, Regs, Mux |

Câu 6 Giai đoạn nhận vào các thao tác, toán hạng và trả về kết quả tính toán thuộc công đoạn nào trong quá trình thực thi lệnh của MIPS? (G1)

| | | | |
|--------|------------------|-----------------------|-----------------|
| A. ALU | B. Memory access | C. Instruction decode | D. Result write |
|--------|------------------|-----------------------|-----------------|

Câu 7 Trình tự thực hiện lệnh của lệnh thuộc nhóm lệnh logic nào là đúng? (G2)

| | |
|----|--|
| A. | Nạp lệnh – sử dụng ALU – đọc thanh ghi – ghi thanh ghi |
| B. | Nạp lệnh – đọc thanh ghi – sử dụng ALU – ghi thanh ghi |
| C. | Sử dụng ALU – nạp lệnh – đọc thanh ghi – ghi thanh ghi |
| D. | Nạp lệnh – đọc thanh ghi – ghi thanh ghi – sử dụng ALU |

Câu 8 Giá trị output của khối “Sign-extend” bằng bao nhiêu khi mã sau được thanh ghi PC trữ tới trong quá trình thực thi: 0x2149ff9c? (G3)

| | | | |
|-----------------|---------------|---------------|---------------|
| A. 0xffffffff9c | B. 0x2149ff9c | C. 0x0000ff9c | D. 0x00002149 |
|-----------------|---------------|---------------|---------------|

Câu 9 Giá trị output của “Read data 1” bằng bao nhiêu khi mã lệnh sau được thanh ghi PC trữ tới trong quá trình thực thi: 0x2149ff9c? Biết giá trị của thanh ghi số 9 và thanh ghi số 10 tương ứng là: 0xffffffffb và 0x0000000b. (G2) [Thanh ghi s 10: 0x0000000b](#)

| | | | |
|-------|------|-------|-------|
| A. 10 | B. 9 | C. 11 | D. -5 |
|-------|------|-------|-------|

Câu 10 Giá trị đầu ra Zero của ALU bằng bao nhiêu khi mã lệnh sau được thanh ghi PC trữ tới trong quá trình thực thi: 0x2149ff90? Biết giá trị của thanh ghi số 9 và thanh ghi số 10 tương ứng là: 0xffffffff0 và 0x0000000c (G3) [0010_0001_0100_1001](#)

| | | | |
|-----------------|----------------|------|------|
| A. 0xffffffff90 | B. 0xffffffffc | C. 0 | D. 1 |
|-----------------|----------------|------|------|

Câu 11 Khối chức năng nào thuộc datapath KHÔNG tham gia vào lệnh lw \$s1, 12(\$t1) (G3)

| | | | |
|----------|----------|-------------------------|--------|
| A. I-mem | B. D-mem | C. Add sau shift left 2 | D. ALU |
|----------|----------|-------------------------|--------|

Câu 12 Trong datapath đã học như hình 1, giá trị của hai tín hiệu điều khiển “Branch” và “Zero” cùng bằng 1 trong trường hợp nào sau đây. Giả sử hiện tại hai thanh ghi \$s1 và \$s2 đều đang bằng 10 (G3)

| | | | |
|-----------------------|-------------------------|-----------------------|----------------------|
| A. beq 10, \$s1, \$s2 | B. add \$s3, \$s1, \$s2 | C. beq \$s1, \$s2, 10 | D. sw \$s1, 10(\$s2) |
|-----------------------|-------------------------|-----------------------|----------------------|

Câu 13 Khối mạch nào trong các khối mạch sau trong datapath là khối mạch tổ hợp (G3)

| | | | |
|----------|-------------|--------|----------|
| A. I-mem | B. Register | C. Add | D. D-mem |
|----------|-------------|--------|----------|

Câu 14 Chu kỳ xung clock cần cho processor đơn chu kỳ cho 8 lệnh đã học theo datapath ở hình 1 là bao nhiêu với dữ liệu cho như bảng sau? 350+40+220+140+40+220+40+300

| I-Mem | Add | Mux | ALU | Regs | D-Mem |
|-------|-------|------|-------|-------|-------|
| 350ps | 100ps | 40ps | 140ps | 220ps | 300ps |

| | | | |
|----------|---------|-----------|-----------|
| A. 350ps | B. 40ps | C. 1150ps | D. 1350ps |
|----------|---------|-----------|-----------|

Câu 15 Cho biết tổng số chu kỳ clock cần thiết để chạy đoạn lệnh bên dưới trong trường hợp KHÔNG dùng kỹ thuật forwarding và có dùng kỹ thuật forwarding (G3)

```
lw $s1, 20($s6)           if id ex m w
sub $t1, $t2, $s2         if id ex m w
sw $s6, 50($t1)           ex m w
```

| | | | |
|-----------|-----------|-----------|------------|
| A. 7 và 9 | B. 8 và 6 | C. 9 và 7 | D. 10 và 8 |
|-----------|-----------|-----------|------------|

Câu 16 Giả sử mỗi công đoạn trong pipeline có thời gian hoạt động như bảng dưới. Chu kỳ xung clock cần cho processor là bao nhiêu nếu processor thiết kế có pipeline và không pipeline (G3)

| | | | | | |
|-------------------|-------|-------|-------|-------|-------|
| Có pipeline: max | IF | ID | EX | MEM | WB |
| Không pipeline: + | 320ps | 420ps | 350ps | 510ps | 120ps |

| | | | |
|-------------------|--------------------|-------------------|--------------------|
| A. 510ps và 740ps | B. 510ps và 1720ps | C. 630ps và 510ps | D. 630ps và 1600ps |
|-------------------|--------------------|-------------------|--------------------|

Câu 17 Giá trị của tín hiệu ALUOp từ khối Control là bao nhiêu khi thực thi lệnh lw rt, offs(rs)? (G3)

| | | | |
|-------|-------|-------|-------|
| A. 00 | B. 01 | C. 10 | D. 11 |
|-------|-------|-------|-------|

Câu 18 Cho \$s1 = 0x00002004; \$s2 = 0x10010004, sau khi thực hiện lệnh “sw \$s1, 4(\$s2)”, giá trị của thanh ghi nào bị thay đổi? (G3)

| | | | |
|---------|---------|------------------------|--------------------------|
| A. \$s1 | B. \$s2 | C. Cả hai đều thay đổi | D. Cả hai không thay đổi |
|---------|---------|------------------------|--------------------------|

Câu 19 Cho lệnh sau: “sw \$s1, 0(\$s2)”, giá trị tín hiệu RegWrite là? (G3)

| | | | |
|------|------|-------|------|
| A. 0 | B. 1 | C. 11 | D. x |
|------|------|-------|------|

Câu 20 Trong các câu lệnh sau, câu lệnh nào có tín hiệu ALUSrc = 1 (G3) sign extend

| | | | |
|-------|--------|-------|--------|
| A. lw | B. beq | C. or | D. add |
|-------|--------|-------|--------|

Câu 21 Giả sử rằng mỗi lệnh cần 5 công đoạn thực hiện và thời gian thực hiện mỗi công đoạn như sau:

| IF | ID | EX | MEM | WB |
|------|------|------|-------|------|
| 20ns | 30ns | 50ns | 120ns | 80ns |

Cho 2 câu lệnh sau:

add \$s1, \$s3, \$s3 **MAX * 6**

add \$s2, \$s1, \$s3

Các câu lệnh trên thực thi trong processor thiết kế pipeline 5 tầng. Hỏi thời gian cần thiết để thực thi cả 2 câu lệnh trên trong trường hợp sử dụng kỹ thuật nhìn trước (forwarding)? (G3)

| | | | |
|--------|---------------|--------|--------|
| A. 660 | B. 720 | C. 780 | D. 840 |
|--------|---------------|--------|--------|

Câu 22 Giả sử rằng mỗi lệnh cần 5 công đoạn thực hiện và thời gian thực hiện mỗi công đoạn như sau:

| IF | ID | EX | MEM | WB |
|------|------|------|-------|------|
| 20ns | 30ns | 50ns | 120ns | 80ns |

Cho 2 câu lệnh sau:

add \$s1, \$s3, \$s3

add \$s2, \$s1, \$s3

MAX * 8

if id ex mem wb

if id ex mem wb

Các câu lệnh trên thực thi trong processor thiết kế pipeline 5 tầng. Hỏi thời gian cần thiết để thực thi cả 2 câu lệnh trên trong trường hợp KHÔNG sử dụng kỹ thuật nhìn trước (forwarding)? (G3)

| | | | |
|--------|---------|---------------|---------|
| A. 880 | B. 1040 | C. 960 | D. 1120 |
|--------|---------|---------------|---------|

Câu 23 Kỹ thuật forwarding có thể hỗ trợ giải quyết xung đột dữ liệu hiệu quả và nó ngăn chặn tất cả các trường hợp chu kỳ rồi (G3)

| | |
|---------|---------------|
| A. Đúng | B. Sai |
|---------|---------------|

Câu 24 Phát biểu nào sau đây **ĐÚNG** khi nói về xung đột cấu trúc (G1)

| | |
|----|---|
| A. | Có hai lệnh cùng truy xuất vào một tài nguyên phần cứng nào đó cùng một lúc |
| B. | Một lệnh không thể thực thi do lệnh nào vào không phải là lệnh được cần |
| C. | Một lệnh cần dữ liệu để xử lý nhưng dữ liệu đó chưa có sẵn |
| D. | Có hai lệnh cùng cần dữ liệu để xử lý nhưng dữ liệu đó chưa có sẵn |

Câu 25 Công đoạn đọc opcode để xác định kiểu lệnh và chiều dài của từng trường trong mã máy và đọc dữ liệu từ các thanh ghi cần thiết là công đoạn nào trong 5 công đoạn thực thi lệnh của kiến trúc MIPS? (G1)

| | |
|----|---------------------------------|
| A. | Công đoạn giải mã |
| B. | Công đoạn tìm nạp các toán hạng |
| C. | Công đoạn tìm nạp lệnh |
| D. | Công đoạn thực thi lệnh |

Câu 26 Giả sử có thiết kế như sau: mỗi lệnh chỉ sử dụng đúng các giai đoạn cần có của nó, có thể lấy nhiều chu kỳ để hoàn thành, nhưng một lệnh phải hoàn thành xong thì những lệnh khác mới được nạp vào. Đây là thiết kế gì? (G1)

| | |
|----|---------------------------|
| A. | Thiết kế đa chu kỳ |
| B. | Thiết kế đơn chu kỳ |
| C. | Thiết kế pipeline |
| D. | Thiết kế đa công đoạn |

Dữ kiện bên dưới sử dụng cho 4 câu hỏi tiếp theo (từ câu 27 đến câu 30)

Cho một bộ xử lý MIPS 32 bits (có datapath và control như hình đã học).

Biết PC = 0x00400000; \$t1 = 0x10010020; \$t3 = 0x00000015; Word nhớ tại địa chỉ 0x10010030 có nội dung/giá trị bằng 0x00000015

Nếu đoạn chương trình sau được thực thi:

0x00400000 *addi \$s0, \$t1, 8* *\$s0 = 0x10010028*

0x00400004 *lw \$t2, 8(\$s0)* *\$t2 = 0x00000015*

0x00400008 *beq \$t3, \$t2, ABC* 0001_0001_0110_1010_0000_0000_0000_0001

0x0040000C *add \$t2, \$t3, \$t4* 0x116A0001

0x00400010 *ABC: sub \$t3, \$t4, \$t5*

Khi bộ xử lý trên đang thực thi vừa xong công đoạn ALU ở câu lệnh thứ ba, trả lời các câu hỏi bên dưới vào bảng trả lời trắc nghiệm ở trang đầu

Câu 27 Ngõ ra của khối Instruction Memory là bao nhiêu? (G2)

| | | | |
|---------------|---------------|---------------|---------------|
| A. 0x116A0010 | B. 0x116A0001 | C. 0x116B0001 | D. 0x116B0010 |
|---------------|---------------|---------------|---------------|

Câu 28 Giá trị của thanh ghi PC là bao nhiêu? (G2)

| | | | |
|---------------|---------------|---------------|---------------|
| A. 0x00400010 | B. 0x0040000C | C. 0x00400008 | D. 0x00400004 |
|---------------|---------------|---------------|---------------|

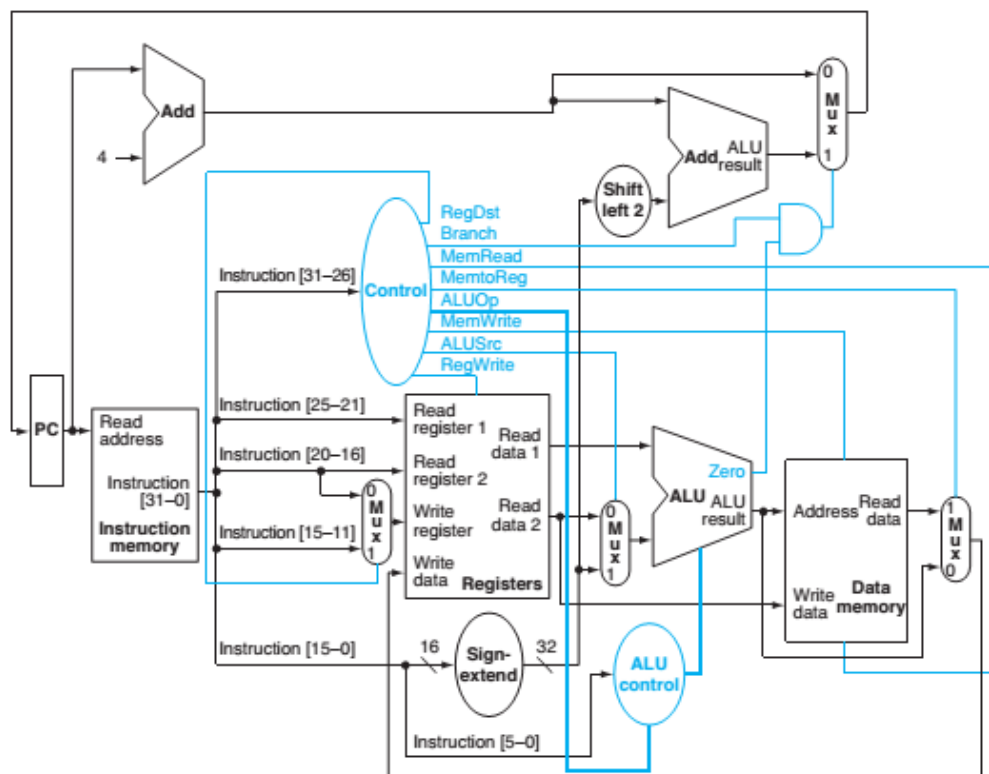
Câu 29 Kết quả tại đầu ra ALU result bằng bao nhiêu? (G2)

| | | | |
|---------------|---------------|---------------|---------------|
| A. 0x00000000 | B. 0x00000030 | C. 0x0000002A | D. 0x00000001 |
|---------------|---------------|---------------|---------------|

Câu 30 Ngõ ra tại khối ALU control là bao nhiêu? (0110)(ghi vào bảng trả lời trắc nghiệm) (G2)

Duyệt đề Khoa/Bộ Môn

Giáo viên ra đề



Hình 1 Sơ đồ datapath

MIPS Reference Data

①



CORE INSTRUCTION SET

| NAME, MNEMONIC | FOR-MAT | OPERATION (in Verilog) | OPCODE / FUNCT (Hex) |
|-----------------------------|---------|--|---------------------------|
| Add | add R | $R[rd] = R[rs] + R[rt]$ | (1) 0 / 20 _{hex} |
| Add Immediate | addi I | $R[rt] = R[rs] + \text{SignExtImm}$ | (1,2) 8 _{hex} |
| Add Imm. Unsigned | addiu I | $R[rt] = R[rs] + \text{SignExtImm}$ | (2) 9 _{hex} |
| Add Unsigned | addu R | $R[rd] = R[rs] + R[rt]$ | 0 / 21 _{hex} |
| And | and R | $R[rd] = R[rs] \& R[rt]$ | 0 / 24 _{hex} |
| And Immediate | andi I | $R[rt] = R[rs] \& \text{ZeroExtImm}$ | (3) C _{hex} |
| Branch On Equal | beq I | if($R[rs] == R[rt]$) $PC = PC + 4 + \text{BranchAddr}$ | (4) 4 _{hex} |
| Branch On Not Equal | bne I | if($R[rs] != R[rt]$) $PC = PC + 4 + \text{BranchAddr}$ | (4) 5 _{hex} |
| Jump | j J | $PC = \text{JumpAddr}$ | (5) 2 _{hex} |
| Jump And Link | jal J | $R[31] = PC + 8; PC = \text{JumpAddr}$ | (5) 3 _{hex} |
| Jump Register | jr R | $PC = R[rs]$ | 0 / 08 _{hex} |
| Load Byte Unsigned | lbu I | $R[rt] = \{24'b0, M[R[rs] + \text{SignExtImm}](7:0)\}$ | (2) 24 _{hex} |
| Load Halfword Unsigned | lhu I | $R[rt] = \{16'b0, M[R[rs] + \text{SignExtImm}](15:0)\}$ | (2) 25 _{hex} |
| Load Linked | ll I | $R[rt] = M[R[rs] + \text{SignExtImm}]$ | (2,7) 30 _{hex} |
| Load Upper Imm. | lui I | $R[rt] = \{\text{imm}, 16'b0\}$ | f _{hex} |
| Load Word | lw I | $R[rt] = M[R[rs] + \text{SignExtImm}]$ | (2) 23 _{hex} |
| Nor | nor R | $R[rd] = \sim (R[rs] R[rt])$ | 0 / 27 _{hex} |
| Or | or R | $R[rd] = R[rs] R[rt]$ | 0 / 25 _{hex} |
| Or Immediate | ori I | $R[rt] = R[rs] \text{ZeroExtImm}$ | (3) d _{hex} |
| Set Less Than | slt R | $R[rd] = (R[rs] < R[rt]) ? 1 : 0$ | 0 / 2a _{hex} |
| Set Less Than Imm. | slti I | $R[rt] = (R[rs] < \text{SignExtImm}) ? 1 : 0$ | (2) a _{hex} |
| Set Less Than Imm. Unsigned | sltiu I | $R[rt] = (R[rs] < \text{SignExtImm}) ? 1 : 0$ | (2,6) b _{hex} |
| Set Less Than Unsig. | sltu R | $R[rd] = (R[rs] < R[rt]) ? 1 : 0$ | (6) 0 / 2b _{hex} |
| Shift Left Logical | sll R | $R[rd] = R[rt] \ll \text{shamt}$ | 0 / 00 _{hex} |
| Shift Right Logical | srl R | $R[rd] = R[rt] \gg \text{shamt}$ | 0 / 02 _{hex} |
| Store Byte | sb I | $M[R[rs] + \text{SignExtImm}](7:0) = R[rt](7:0)$ | (2) 28 _{hex} |
| Store Conditional | sc I | $M[R[rs] + \text{SignExtImm}] = R[rt];$ $R[rt] = (\text{atomic}) ? 1 : 0$ | (2,7) 38 _{hex} |
| Store Halfword | sh I | $M[R[rs] + \text{SignExtImm}](15:0) = R[rt](15:0)$ | (2) 29 _{hex} |
| Store Word | sw I | $M[R[rs] + \text{SignExtImm}] = R[rt]$ | (2) 2b _{hex} |
| Subtract | sub R | $R[rd] = R[rs] - R[rt]$ | (1) 0 / 22 _{hex} |
| Subtract Unsigned | subu R | $R[rd] = R[rs] - R[rt]$ | 0 / 23 _{hex} |

- (1) May cause overflow exception
- (2) $\text{SignExtImm} = \{16\{\text{immediate}[15]\}, \text{immediate}\}$
- (3) $\text{ZeroExtImm} = \{16\{1b'0\}, \text{immediate}\}$
- (4) $\text{BranchAddr} = \{14\{\text{immediate}[15]\}, \text{immediate}, 2'b0\}$
- (5) $\text{JumpAddr} = \{PC + 4[31:28], \text{address}, 2'b0\}$
- (6) Operands considered unsigned numbers (vs. 2's comp.)
- (7) Atomic test&set pair; $R[rt] = 1$ if pair atomic, 0 if not atomic

BASIC INSTRUCTION FORMATS

| R | opcode | rs | rt | rd | shamt | funct |
|---|--------|---------|-------|-----------|-------|-------|
| | 31 | 26 25 | 21 20 | 16 15 | 11 10 | 6 5 |
| | 0 | | | | | |
| I | opcode | rs | rt | immediate | | |
| | 31 | 26 25 | 21 20 | 16 15 | | |
| | 0 | | | | | |
| J | opcode | address | | | | |
| | 31 | 26 25 | | | | |
| | 0 | | | | | |

ARITHMETIC CORE INSTRUCTION SET

| NAME, MNEMONIC | FOR-MAT | OPERATION | OPCODE / FUNCT (Hex) |
|--|-----------|---|----------------------|
| Branch On FP True | bclt FI | if($FPcond$) $PC = PC + 4 + \text{BranchAddr}$ | (4) 11/8/1/-- |
| Branch On FP False | bclt FI | if(! $FPcond$) $PC = PC + 4 + \text{BranchAddr}$ | (4) 11/8/0/-- |
| Divide | div R | $Lo = R[rs] / R[rt]; Hi = R[rs] \% R[rt]$ | 0/--/--/1a |
| Divide Unsigned | divu R | $Lo = R[rs] / R[rt]; Hi = R[rs] \% R[rt]$ | (6) 0/--/--/1b |
| FP Add Single | add.s FR | $F[fd] = F[fs] + F[ft]$ | 11/10/--/0 |
| FP Add Double | add.d FR | $\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} + \{F[ft], F[ft+1]\}$ | 11/11/--/0 |
| FP Compare Single | c.x.s* FR | $FPcond = (F[fs] \text{ op } F[ft]) ? 1 : 0$ | 11/10/--/y |
| FP Compare Double | c.x.d* FR | $FPcond = (\{F[fs], F[fs+1]\} \text{ op } \{F[ft], F[ft+1]\}) ? 1 : 0$ | 11/11/--/y |
| * (x is eq, lt, or le) (op is ==, <, or <=) (y is 32, 3c, or 3e) | | | |
| FP Divide Single | div.s FR | $F[fd] = F[fs] / F[ft]$ | 11/10/--/3 |
| FP Divide Double | div.d FR | $\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} / \{F[ft], F[ft+1]\}$ | 11/11/--/3 |
| FP Multiply Single | mul.s FR | $F[fd] = F[fs] * F[ft]$ | 11/10/--/2 |
| FP Multiply Double | mul.d FR | $\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} * \{F[ft], F[ft+1]\}$ | 11/11/--/2 |
| FP Subtract Single | sub.s FR | $F[fd] = F[fs] - F[ft]$ | 11/10/--/1 |
| FP Subtract Double | sub.d FR | $\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} - \{F[ft], F[ft+1]\}$ | 11/11/--/1 |
| Load FP Single | lwc1 I | $F[rt] = M[R[rs] + \text{SignExtImm}]$ | (2) 31/--/--/1 |
| Load FP Double | ldc1 I | $F[rt+1] = M[R[rs] + \text{SignExtImm} + 4]$ | (2) 35/--/--/1 |
| Move From Hi | mghi R | $R[rd] = Hi$ | 0 / --/--/10 |
| Move From Lo | mflr R | $R[rd] = Lo$ | 0 / --/--/12 |
| Move From Control | mfc0 R | $R[rd] = CR[rs]$ | 10 / 0/--/0 |
| Multiply | mult R | $\{Hi, Lo\} = R[rs] * R[rt]$ | 0/--/--/18 |
| Multiply Unsigned | multu R | $\{Hi, Lo\} = R[rs] * R[rt]$ | (6) 0/--/--/19 |
| Shift Right Arith. | sra R | $R[rd] = R[rt] \gg \text{shamt}$ | 0/--/--/3 |
| Store FP Single | swc1 I | $M[R[rs] + \text{SignExtImm}] = F[rt]$ | (2) 39/--/--/1 |
| Store FP Double | sdc1 I | $M[R[rs] + \text{SignExtImm}] = F[rt];$ $M[R[rs] + \text{SignExtImm} + 4] = F[rt+1]$ | (2) 3d/--/--/1 |

FLOATING-POINT INSTRUCTION FORMATS

| FR | opcode | fmt | ft | fs | fd | funct |
|----|--------|-------|-------|-----------|-------|-------|
| | 31 | 26 25 | 21 20 | 16 15 | 11 10 | 6 5 |
| | 0 | | | | | |
| FI | opcode | fmt | ft | immediate | | |
| | 31 | 26 25 | 21 20 | 16 15 | | |
| | 0 | | | | | |

PSEUDOINSTRUCTION SET

| NAME | MNEMONIC | OPERATION |
|------------------------------|----------|--|
| Branch Less Than | blt | if($R[rs] < R[rt]$) $PC = \text{Label}$ |
| Branch Greater Than | bgt | if($R[rs] > R[rt]$) $PC = \text{Label}$ |
| Branch Less Than or Equal | bte | if($R[rs] \leq R[rt]$) $PC = \text{Label}$ |
| Branch Greater Than or Equal | bge | if($R[rs] \geq R[rt]$) $PC = \text{Label}$ |
| Load Immediate | li | $R[rd] = \text{immediate}$ |
| Move | move | $R[rd] = R[rs]$ |

REGISTER NAME, NUMBER, USE, CALL CONVENTION

| NAME | NUMBER | USE | PRESERVED ACROSS A CALL? |
|-----------|--------|---|--------------------------|
| \$zero | 0 | The Constant Value 0 | N.A. |
| \$at | 1 | Assembler Temporary | No |
| \$v0-\$v1 | 2-3 | Values for Function Results and Expression Evaluation | No |
| \$a0-\$a3 | 4-7 | Arguments | No |
| \$t0-\$t7 | 8-15 | Temporaries | No |
| \$s0-\$s7 | 16-23 | Saved Temporaries | Yes |
| \$t8-\$t9 | 24-25 | Temporaries | No |
| \$k0-\$k1 | 26-27 | Reserved for OS Kernel | No |
| \$gp | 28 | Global Pointer | Yes |
| \$sp | 29 | Stack Pointer | Yes |
| \$fp | 30 | Frame Pointer | Yes |
| \$ra | 31 | Return Address | Yes |

Copyright 2009 by Elsevier, Inc., All rights reserved. From Patterson and Hennessy, *Computer Organization and Design*, 4th ed.

Hình 2 Bảng tham khảo kiến trúc tập lệnh MIPS