

BÀI GIẢI BÀI TẬP 2

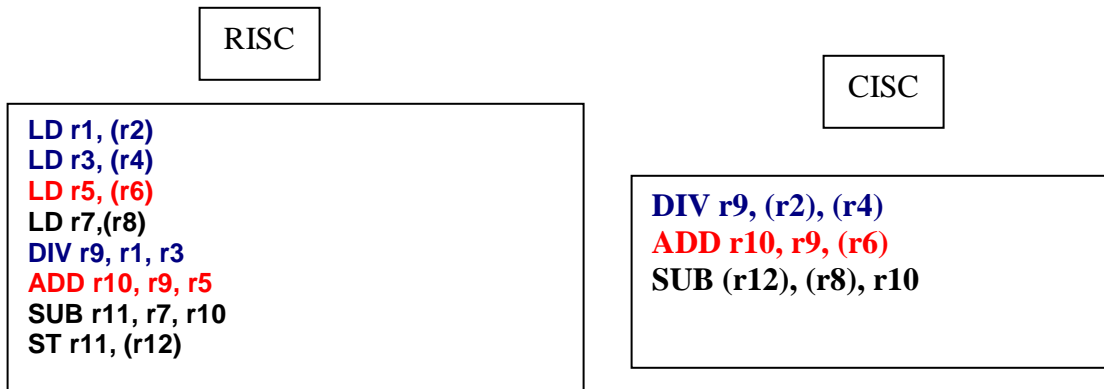
Load R15, (1024)																Mã tác vụ lệnh Load: 001001															
d ₃₁	d ₃₀	d ₂₉	d ₂₈	d ₂₇	d ₂₆	d ₂₅	d ₂₄	d ₂₃	d ₂₂	d ₂₁	d ₂₀	d ₁₉	d ₁₈	d ₁₇	d ₁₆	d ₁₅	d ₁₄	d ₁₃	d ₁₂	d ₁₁	d ₁₀	d ₉	d ₈	d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀
0	0	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Opcode						Th/g đích					Th/g nguồn					Do doi 16 bit															
Load R.20, 200(R12)																Mã tác vụ lệnh Load: 001001															
d ₃₁	d ₃₀	d ₂₉	d ₂₈	d ₂₇	d ₂₆	d ₂₅	d ₂₄	d ₂₃	d ₂₂	d ₂₁	d ₂₀	d ₁₉	d ₁₈	d ₁₇	d ₁₆	d ₁₅	d ₁₄	d ₁₃	d ₁₂	d ₁₁	d ₁₀	d ₉	d ₈	d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀
0	0	1	0	0	1	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0
Opcode						Th/g Số Liệu					Th/g Địa chỉ					Do doi 16 bit															
Store R14, (R22)																Mã tác vụ lệnh Store: 001010															
d ₃₁	d ₃₀	d ₂₉	d ₂₈	d ₂₇	d ₂₆	d ₂₅	d ₂₄	d ₂₃	d ₂₂	d ₂₁	d ₂₀	d ₁₉	d ₁₈	d ₁₇	d ₁₆	d ₁₅	d ₁₄	d ₁₃	d ₁₂	d ₁₁	d ₁₀	d ₉	d ₈	d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀
0	0	1	0	1	0	0	1	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Opcode						Th/g SL					Th/g DC					Do doi 16 bit															
Add R4, R13, R12																Mã tác vụ lệnh Add: 011010															
d ₃₁	d ₃₀	d ₂₉	d ₂₈	d ₂₇	d ₂₆	d ₂₅	d ₂₄	d ₂₃	d ₂₂	d ₂₁	d ₂₀	d ₁₉	d ₁₈	d ₁₇	d ₁₆	d ₁₅	d ₁₄	d ₁₃	d ₁₂	d ₁₁	d ₁₀	d ₉	d ₈	d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀
0	1	1	0	1	0	0	0	1	0	0	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Opcode						Th/g Dich					Th/g Nguồn 1					Th/g Nguồn 2					Opcode mở rộng						1				
Sub R3, R17, # -1000																Mã tác vụ lệnh Sub: 011011															
d ₃₁	d ₃₀	d ₂₉	d ₂₈	d ₂₇	d ₂₆	d ₂₅	d ₂₄	d ₂₃	d ₂₂	d ₂₁	d ₂₀	d ₁₉	d ₁₈	d ₁₇	d ₁₆	d ₁₅	d ₁₄	d ₁₃	d ₁₂	d ₁₁	d ₁₀	d ₉	d ₈	d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀
0	1	1	0	1	1	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	0	0
Opcode						Th/g Dich					Th/g Nguồn 1					Số cơ sở 16 bit															

Tham khảo

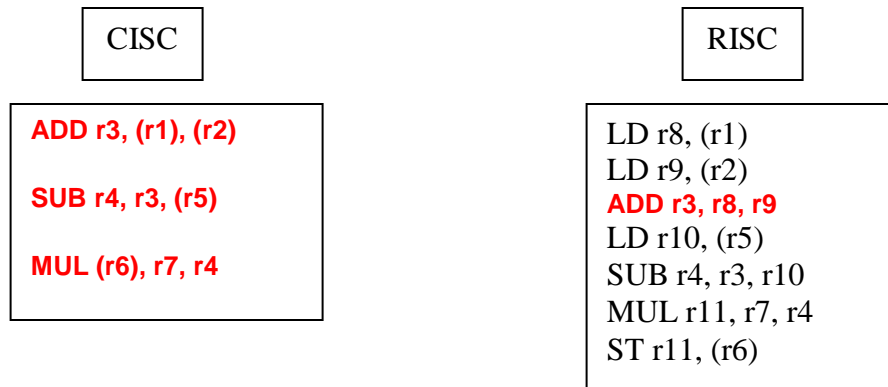
n	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2 ⁿ	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

6. Nguyên tắc: Để tăng tốc độ chạy chương trình, kiến trúc RISC hạn chế thâm nhập bộ nhớ. Vì vậy, kiến trúc RISC trang bị nhiều thanh ghi và qui định các lệnh ALU chỉ sử dụng toán hạng thanh ghi, nếu cần thực hiện phép tính với toán hạng ở bộ nhớ thì phải tải về thanh ghi với lệnh Load, kết quả của phép tính cũng chỉ được lưu vào thanh ghi, nếu cần thì dùng lệnh Store để ghi vào bộ nhớ. Ngược lại, kiến trúc CISC thì có thể tính trực tiếp với toán hạng bộ nhớ (kể cả toán hạng nguồn và toán hạng đích).

Trong bài giải sau đây, các lệnh cùng màu trong kiến trúc RISC tương đương với lệnh cùng màu đó trong kiến trúc CISC.



7.



8. Trên một hệ thống với từ dữ liệu 8 bit, giả sử số liệu trên thanh ghi A là N, cho biết kết quả:

- a. Lệnh dịch trái logic 2 bit trên thanh ghi A, với N = 25. Nhận xét.
- b. Lệnh dịch phải logic 2 bit trên thanh ghi A, với N = 25. Nhận xét.
- c. Lệnh dịch trái số học 3 bit trên thanh ghi A, với N = -15. Nhận xét.
- d. Lệnh dịch trái số học 3 bit trên thanh ghi A, với N = -17. Nhận xét.
- e. Lệnh dịch phải số học 2 bit trên thanh ghi A, với N = 16. Nhận xét.
- f. Lệnh dịch phải số học 2 bit trên thanh ghi A, với N = 17. Nhận xét.

Giải:

a. Dịch trái logic: N=25, m=2

Trước khi dịch 25

0	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---

Sau khi dịch 100

0	1	1	0	0	1	0	0
---	---	---	---	---	---	---	---

Nhận xét:

$100 = 25 \times 2^2 \rightarrow$ phép **dịch trái logic** 2 bit một số dương (không có dấu) tương đương với phép nhân cho 2^2 .

Tổng quát, phép dịch trái logic m bit một số dương (không có dấu) N tương đương với phép nhân N cho 2^m .

Tuy nhiên, một số không có dấu n bit chỉ biểu diễn được con số dương lớn nhất là $(2^n - 1)$. Vì vậy, kết luận trên chỉ đúng thì $N \times 2^m \leq (2^n - 1)$, nghĩa là tùy thuộc vào N, mà chọn số bit dịch m thích hợp. Trong bài tập này $N = 25$, nên m lớn nhất là 3.

b. Dịch phải logic: $N=25, m=2$



Trước khi dịch 25

25

0	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---

Sau khi dịch

198 (số không có dấu)

1	1	0	0	0	1	1	0
---	---	---	---	---	---	---	---

Nhận xét:

Phép dịch phải logic m bit một số dương (số không có dấu) **không** có quan hệ với phép nhân cho 2^m .

Lưu ý: Có một số giáo trình qui định dịch phải logic như sau:



Trước khi dịch 25

25

0	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---

Sau khi dịch

6 (số không có dấu)

0	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---

Nhận xét:

Phép dịch phải logic m bit một số dương (số không có dấu) tương đương với phép chia cho 2^m (hay nhân cho 2^{-m}) lấy phần nguyên.

Qui định bit 0 hay bit 1 nhảy vào là tùy thuộc vào CPU, khi thì ta chọn trường hợp bit 0 nhảy vào.

c. Dịch trái số học: $N = -15, m = 3$

Trước khi dịch

-15 (số bù 2)

1	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---

Sau khi dịch

-120 (số bù 2)

1	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---

Nhận xét:

$-120 = -15 \times 2^3 \rightarrow$ phép **dịch trái đại** số 3 bit một số **âm** (số có dấu kiểu bù 2) tương đương với phép nhân cho 2^3 .

Tổng quát, phép dịch trái **đại số** m bit một số **số âm** (số có dấu kiểu bù 2) N tương đương với phép nhân N cho 2^m .

Tuy nhiên, một số không có dấu n bit chỉ biểu diễn được con số âm nhất là: $-(2^{n-1}-1)$. Vì vậy, kết luận trên chỉ đúng thì $N \times 2^m \geq -(2^{n-1}-1)$, nghĩa là tùy thuộc vào N , mà chọn số bit dịch m thích hợp. Trong bài tập này $N = -15$, $N = 8$, số âm nhất là -127 , nên m lớn nhất là 3.

Kết luận:

- Dịch logic dùng để thực hiện phép nhân (dịch trái)/chia (dịch phải, trường hợp bit 0 nhảy vào) một số nguyên dương (không có dấu) cho lũy thừa của 2.
- Dịch số học dùng để thực hiện phép nhân (dịch trái)/chia (dịch phải, trường hợp bit 0 nhảy vào) một số nguyên dương (không có dấu) cho lũy thừa của 2.

d. Dịch phải số học: $N=-15$, $m=3$

Trước khi dịch	-15 (số bù 2)	<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	0
1	1	1	1	0	0	0	1											
1	1	1	1	1	1	1	0											
Sau khi dịch	-2 (số bù 2)																	

Nhận xét:

Phép dịch phải đại số n bit một số âm **không** có quan hệ với phép nhân cho 2^n .

e. Dịch phải số học: $N=16$, $m=2$

Trước khi dịch	16 (số dương)	<table><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	0	0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0										
0	0	0	0	0	1	0										
Sau khi dịch	4 (số dương)															

Nhận xét:

$4 = 16 \times 2^{-2} \rightarrow$ phép **dịch phải đại số** 2 bit một số **dương** tương đương với phép nhân cho 2^{-2} (chia cho 2^2).

f. Dịch trái số học: $N=16$, $m=2$

Qui tắc của dịch trái số học như sau:



Trước khi dịch	16 (số dương)	<table><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0											
0	1	0	0	0	0	0	0											
Sau khi dịch	64 (số dương)																	

Nhận xét:

$64 = 16 \times 2^2 \rightarrow$ phép **dịch trái** số học 2 bit một số **dương** tương đương với phép nhân cho 2^2 .

Tổng quát, phép dịch phải **số học** m bit một số **số dương** N tương đương với phép chia N cho 2^m lấy phần nguyên. Tuy nhiên, m phải thỏa điều kiện: $2^m \leq N$

Phép dịch trái **số học** m bit một số **số dương** N tương đương với nhân N cho 2^m .

Tuy nhiên, một số có dấu n bit chỉ biểu diễn được con số dương lớn nhất là $(2^{n-1} - 1)$. Vì vậy, kết luận trên chỉ đúng thì $N \times 2^m \leq (2^{n-1} - 1)$, nghĩa là tùy thuộc vào N, mà chọn số bit dịch m thích hợp. Trong bài tập này $N = 16$, nên m lớn nhất là 2.

Lưu ý:

Dịch trái số học: $N=112, m=2$

Trước khi dịch	112 (số dương)	0	1	1	1	0	0	0
Sau khi dịch	64 (số dương)	0	1	0	0	0	0	0

Nhận xét: Ta thấy kết quả dịch trái số học 2 bit thanh ghi có giá trị ban đầu là số 112 và 16 giống nhau (đều bằng 64).

$112 \times 2^2 = 448$ không thỏa điều kiện $N \times 2^m \leq (2^{n-1} - 1) = 127$ (với $n=8$ bit), vì vậy phép **dịch trái số học 2 bit một số dương** trong trường hợp này **không** tương đương với phép nhân cho 2^2 , vì đã bị tràn thanh ghi.

Một số trường hợp sau đây cũng cho cùng kết quả:

Dịch trái số học: $N=16, m=2$

Trước khi dịch	80 (số dương)	0	1	0	1	0	0	0
Sau khi dịch	64 (số dương)	0	1	0	0	0	0	0

Dịch trái số học: $N=16, m=2$

Trước khi dịch	48 (số dương)	0	0	1	1	0	0	0
Sau khi dịch	64 (số dương)	0	1	0	0	0	0	0

9. Hãy hoàn thành bảng trạng thái sau đây của phép toán sau SUBCC Ri,Rj (phép trừ, lưu các bit trạng thái vào thanh ghi trạng thái):

Kết quả	S	Z
=	x	1
>	0	0
<	1	X
>=	0	X
<=	1	0
	0	1

Trong các trường hợp trên, S và Z có thể lấy giá trị là 0 hoặc 1, trường hợp không cần xét bit trạng thái đó thì ghi dấu x.

Bài tập 10:

Kiến trúc RISC dùng thanh ghi R31 để lưu địa chỉ trở về và thanh ghi R30 làm con trỏ ngăn xếp. Xét trường hợp từ chương trình chính gọi thủ tục P1, rồi từ thủ tục P1 gọi thủ tục P2 một lần, sau đó trở về thủ tục P1, rồi trở về chương trình chính. Cho biết lệnh JMPL Ri là lệnh gọi thủ tục có địa chỉ đầu tiên nằm trong Ri, lệnh JMP R31 là lệnh kết thúc thủ tục trở về chương trình gọi thủ tục. Giả sử thủ tục P1 bắt đầu từ địa chỉ ghi trong R1 và thủ tục P2 bắt đầu từ địa chỉ ghi trong R2. Hãy mô tả chương trình thực thi các tác vụ này.

Giải:

JMPL R1 // Gồm các thao tác: R31 := PC; PC := R1//

Gọi thủ tục P2:

ADDI R30, R30, 4

STORE R31, (R30)

JMPL R2 // Gồm các thao tác: R31 := PC; PC := R2//

Trở về thủ tục P1:

JMP R31

Trở về chương trình chính:

LOAD R31, (R30)

SUBI R30, R30, 4

JMP R31