

Bài tập/Thực hành 3

CHƯƠNG 2: KIẾN TRÚC TẬP LỆNH MIPS: CÁC LỆNH ĐIỀU KHIỂN

Danh sách thành viên:

1. Hồ Minh Hưng 2211361
2. Tô Thế Hưng 2211384
3. Nguyễn Lê Gia Kiệt 2211761

Làm thêm Lab 2.2:

1. ENDIANESS.

Cho mảng số nguyên bên dưới.

`.data`

`intArray: .word 0xCA002019, 0xC0002009`

`.text`

`la $a0, intArray`

`lb $t0, 0($a0)`

`lb $t1, 1($a0)`

`lb $t2, 2($a0)`

`lb $t3, 3($a0)`

`lbu $t4, 0($a0)`

`lbu $t5, 1($a0)`

`lbu $t6, 2($a0)`

`lbu $t7, 3($a0)`

(a) Giả sử MIPS được thiết kế theo kiểu BIG ENDIAN, xác định giá trị các ô nhớ (theo byte) của mảng

trên.

(b) Giả sử MIPS được thiết kế theo kiểu LITTLE ENDIAN, xác định giá trị các ô nhớ (theo byte) của mảng trên.

(c) Xác định giá trị các thanh ghi \$t của đoạn code bên dưới, giả sử MIPS được thiết kế theo kiểu BIG ENDIAN.

(d) Xác định giá trị các thanh ghi \$t của đoạn code bên dưới, giả sử MIPS được thiết kế theo kiểu LITTLE ENDIAN.

Giải:

(a)

Địa chỉ 0(\$a0): **0xCA** (tương ứng với \$t0 và \$t4)

Địa chỉ 1(\$a0): **0x00** (tương ứng với \$t1 và \$t5)

Địa chỉ 2(\$a0): **0x20** (tương ứng với \$t2 và \$t6)

Địa chỉ 3(\$a0): **0x19** (tương ứng với \$t3 và \$t7)

(b)

Địa chỉ 0(\$a0): **0x19** (tương ứng với \$t3 và \$t7)

Địa chỉ 1(\$a0): **0x20** (tương ứng với \$t2 và \$t6)

Địa chỉ 2(\$a0): **0x00** (tương ứng với \$t1 và \$t5)

Địa chỉ 3(\$a0): **0xCA** (tương ứng với \$t0 và \$t4)

(c)

\$t0: 0xFFFFFFFFCA (đọc byte đầu tiên, ký tự mở rộng vì lb)

\$t1: 0x00

\$t2: 0x20

\$t3: 0x19

\$t4: 0xCA (ký tự mở rộng vì lbu)

\$t5: 0x00

\$t6: 0x20

\$t7: 0x19

(d)

Đối với kiểu LITTLE ENDIAN, giá trị của các thanh ghi \$t sẽ là:

\$t0: 0x19

\$t1: 0x20

\$t2: 0x00

\$t3: 0xFFFFFFFFCA

\$t4: 0x19

\$t5: 0x20

\$t6: 0x00

\$t7: 0xCA

2. Memory alignment.

Cho đoạn code mips bên dưới

.data

int_1: .word 0xCA002018

char_1: .byte 0xFF

int_2: .word 2018

char_2: .byte 0xCA 0xFE 0xED

.text

la \$a0, int_1

lw \$t0, 0(\$a0)

lw \$t1, 1(\$a0)

lh \$t2, 2(\$a0)

lh \$t3, 3(\$a0)

lb \$t4, 0(\$a0)

lb \$t5, 1(\$a0)

(a) Xác định nội dung của vùng nhớ dữ liệu và xác định các lệnh sẽ gây ra lỗi khi thực thi, giải thích.

Biết MIPS chuẩn được thiết kế theo kiểu BIG ENDIAN.

(b) Xếp lại dữ liệu sao cho bộ nhớ tối ưu hơn (trong kiến trúc 32 bit).

Giải:

(a)

Lỗi có thể xảy ra tại các lệnh lb \$t4, 0(\$a0) và lb \$t5, 1(\$a0) khi load byte từ địa chỉ int_1 và int_1 + 1, vì giá trị sẽ bị mở rộng từ 8-bit signed, gây ra sự hiểu lầm về giá trị thực tế của byte.

(b)

Có thể sắp xếp lại dữ liệu như sau:

.data

int_1: .word 0xCA002018

int_2: .word 2018

char_1: .byte 0xFF

char_2: .byte 0xCA, 0xFE, 0xED