

BÀI TẬP CHƯƠNG 2: ASSEMBLY MIPS VÀ MÃ MÁY

DẠNG 1: CHUYỂN ĐỔI GIỮA MÃ MÁY VÀ ASSEMBLY

Bài tập 1: Chuyển đổi Assembly MIPS sang mã máy (Hex)

- a) add \$t1, \$t2, \$t3
- b) sub \$t0, \$s0, \$s1
- c) addi \$s1, \$s2, 20
- d) lw \$s0, 24(\$t1)
- e) sw \$s3, 40(\$t2)
- f) lw \$t2, -8(\$s0)
- g) sh \$s2, 12(\$t4)
- h) mult \$t5, \$t6
- i) mfhi \$s5
- j) sra \$t1, \$t0, 2
- k) lui \$s6, 12
- l) j Label (Label = 0x00400020)
- m) bne \$t1, \$t2, Label (Label = 0x00400008, PC = 0x00400004)
- n) start: addi \$t0, \$zero, 5
 lw \$t1, 12(\$s0)
 sub \$s1, \$t0, \$t2
 beq \$s1, \$t0, end (1)
 sw \$t2, 20(\$s0)
end: ori \$t3, \$zero, 15
- o) Loop: addi \$s2, \$s2, 2
 sub \$s3, \$s3, \$s4
 beq \$s2, \$s1, Loop (2)
 ori \$s5, \$zero, 12

Chuyển lệnh (2) sang mã máy

Bài tập 2: Chuyển mã máy (Hex) sang assembly MIPS

- a) 0xAD310018
- b) 0x8D28000C
- c) 0x0170001A
- d) 0x02325820
- e) 0x02538820
- f) 0x001090C3
- g) 0x014B4822
- h) 0x85680004
- i) 0x2151fff8
- j) 0x2109FFFC
- k) 0x8e28fff0
- l) 0xAE28FFFC
- m) 0x01534822
- n) 0x08100009 (PC = 0x90400018)
- o) 0x1109fffd
- p) 0x1632fff9

DẠNG 2: CHUYỂN TỪ C SANG ASSEMBLY MIPS VÀ NGƯỢC LẠI

Giả sử f, g, h, i, j = \$s0, \$s1, \$s2, \$s3, \$s4; A, B = \$s6, \$s7

a, b, c, d = \$a0, \$a1, \$a2, \$a3

Bài tập 1: Chuyển từ C sang Assembly MIPS (biết mỗi word có 4 bytes)

- a) $B[5] = A[i] - A[j];$
- b) $B[j] = A[i] + B[2];$
- c) $A[7] = B[i + j];$
- d)

```
i = 5;
if (i < j) {
    f = g - h;
} else {
    f = g + h;
}
```

e)

```
i = j + 2;
if (i == g) {
    f = h + g;
}
```

f)

```
i = 0;
while (i < 5) {
    f = f + g;
    i = i + 1;
}
```

g)

```
for (i = 0; i < 3; i++) {
    f = f + h;
}
```

h)

```
for (i = 0; i < j; i++) {
    f = f * g;
}
```

i)

```
i = 0;
while (i < 10) {
    if (i % 2 == 0) {
        f = f + g;
    } else {
        f = f - h;
    }
    i++;
}
```

j)

```
i = 5;
while (i < 15) {
    f = g << 2;
    h = h + f;
    i++;
}
```

k)

```
i = 0;
do {
    f = f + g;
    i++;
} while (i < j);
```

l)

```
A[3] = f + g;
if (A[3] > h) {
    B[2] = A[3] - h;
}
```

m)

```
B[5] = A[i] + A[j];
if (B[5] < h) {
    f = g + h;
} else {
    f = g - h;
}
```

o)

```
i = 5;
do {
    i = i - 2;
} while (i > 0);
```

p)

```
if (A[i] == g) {
    f = g + h;
} else {
    f = g - h;
}
```

q)

```
if (i >= j) {
    f = A[j] - g;
} else {
    f = g | h;
}
```

r)

```
if (i > 0 && i < 10) {
    f = g + h;
} else {
    f = g - h;
}
```

s)

```
if (i < 5 || i > 20) {
    f = g & h;
} else {
    f = g | h;
}
```

t)

```
int tinh(int a, int b) {
    return a * b;
}
int f = tinh(2, 3);
```

u)

```
int tong(int a, int b) {
    return a + b;
}
f = tong(i, j) + tong(i, 4);
```

v)

```
int tong(int a, int b) {
    return a + b;
}
int func(int a, int b, int c, int d) {
    return tong(a, b) - tong(c, d);
}
```

w)

```
int is_between(int a, int b, int c) {
    return (a <= b && b <= c);
}
```

x)

```
int tong(int a, int b) {
    return a + b;
}
int func(int a, int b, int c, int d) {
    return tong(a, b) + tong(c, d);
}
```

y)

```
int complex_func(int a, int b, int c, int d) {
    int sum1 = a + b;
    int sum2 = c + d;
    return sum1 * sum2;
}
```

z)

```
int TowerHanoi(int N){
    if(N <= 0) return 0;
    else if(N == 1) return 1;
    return 2*TowerHanoi(N-1) + 1;
}
```

Bài tập 2: Chuyển từ Assembly MIPS sang C

Giả sử $f, g, h, i, j = \$s0, \$s1, \$s2, \$s3, \$s4$; $A, B = \$s6, \$s7$

$a, b, c, d = \$a0, \$a1, \$a2, \$a3$

```

a)      beq $s1, $s2, label
        bgt $s2, $s3, exit
        ble $s3, $s4, exit
label:   add $s4, $s4, $s1

b)      li $s3, 10
        ori $t0, $zero, 1
do_loop_b:
        srl $s3, $s3, 2
        bgt $s3, $t0, do_loop_b

c)      sll $t0, $s3, 2
        add $t0, $t0, $s7
        sw $t1, 0($t0)
        bne $t1, $s4, else
        mul $s2, $s0, $s1    # h = f * g
        j end_if_c

else:
        add $s0, $s1, $s2    # f = g + h
end_if_c:

d)      bge $s3, $s4, else    # if i < j
        sll $t0, $s4, 2
        add $t0, $t0, $s7
        sw $t1, 0($t0)
        sub $s0, $t1, $s1    # f = B[j] - g
        j end_if_d

else:
        and $s0, $s1, $s2    # f = g & h
end_if_d:

e)      ori $t0, $zero, 5
        ori $t1, $zero, 15
        ble $s3, $t0, case_1 # if i <= 5
        bge $s3, $t1, case_2 # if i >= 15
        mul $s0, $s1, $s2    # f = g * h
        j end_if_z

case_1:
case_2:
        add $s0, $s1, $s2    # f = g + h
end_if_z:

```

Donate:
TRAN MINH PHU

