TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN – ĐHQG HCM

KHOA KỸ THUẬT MÁY TÍNH

TỔ CHỨC VÀ CẤU TRÚC MÁY TÍNH 2

CHƯƠNG 7
BIÊN DỊCH CHƯƠNG TRÌNH

PHAN ĐÌNH DUY

TP. Hồ Chí Minh, ngày 05 tháng 9 năm 2022

FACULTY OF COMPUTER ENGINEERING

MỤC TIÊU CHƯƠNG

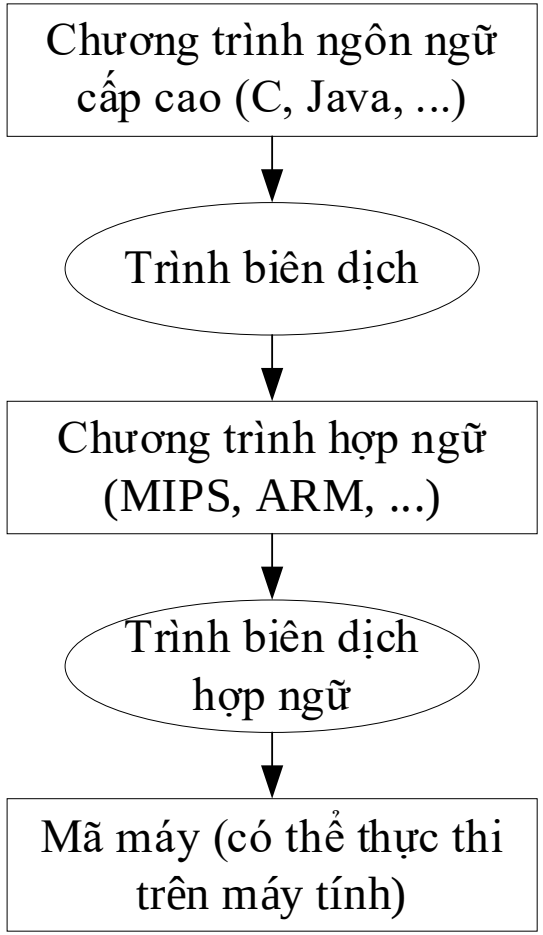
- Biết được khái niệm trình biên dịch và trình thông dịch
- Hiểu được các chương trình biên dịch từ ngôn ngữ cấp cao sang ASM và mã máy
- Hiểu được các biên dịch ngược từ mã máy sang ASM và ngôn ngữ cấp cao

NỘI DUNG

- Trình biên dịch (Compiler)
- Trình biên dịch hợp ngữ - trình thông dịch (Assembler)
- Biên dịch ngược (Reverse-Engineering)
- Bài tập

Trình biên dịch

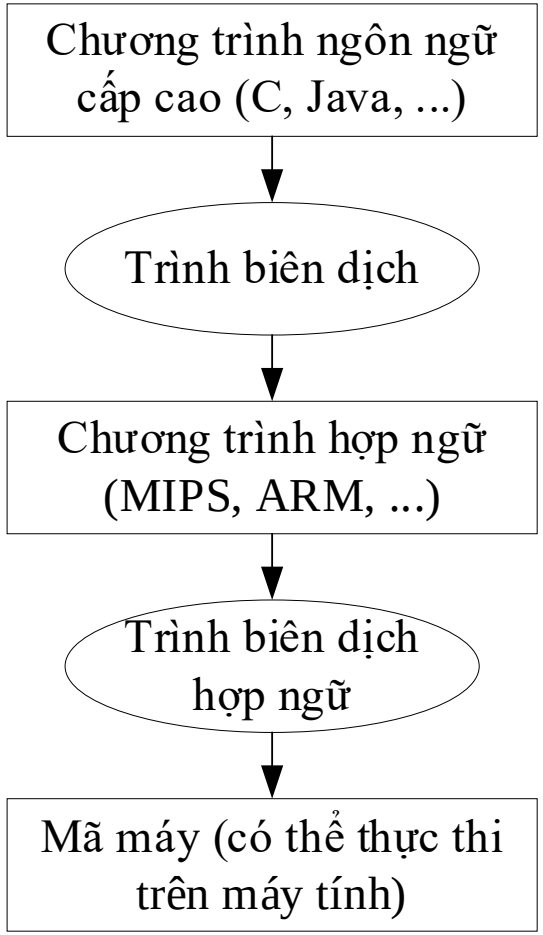
- **Trình biên dịch** có chức năng chuyển chương trình được viết bởi ngôn ngữ lập trình cấp cao thành chương trình hợp ngữ:
  - Ngôn ngữ lập trình cấp cao (C, Java, …) gần với suy nghĩ con người và độc lập phần cứng
  - Hợp ngữ (MIPS, ARM, ...) là một ngôn ngữ gợi nhớ của mã máy, phụ thuộc phần cứng



Trình biên dịch (2) - ví dụ

```
if(a == b)
  c = 2;
else
  c = -1;
d = a + c;
```

```
bne  $a0, $a1, ELSE
addi  $s0, $0, 2
j       ENDIF
ELSE:
        addi  $s0, $0, -1
ENDIF:
        add  $s1, $a0, $s0
```



Trình biên dịch (3) - Quiz 1

- Biên dịch chương trình được viết bằng ngôn ngữ C sau sang hợp ngữ MIPS. Với arraylength và i tưng ứng với thanh ghi $s0 và $s1; mảng arrayvalue có địa chỉ base nằm trong thanh ghi $s5:

```
int arraylength = 5;
for(int i = 0; i < arraylength; i++)
{
    arrayvalue[i] = i;
}
```

Trình biên dịch (3) - Quiz 1

- Biên dịch chương trình được viết bằng ngôn ngữ C sau sang hợp ngữ MIPS. Với arraylength và i tưng ứng với thanh ghi $s0 và $s1; mảng arrayvalue có địa chỉ base nằm trong thanh ghi $s5:

```
int arraylength = 5;          addi $s0, $zero, 5
for(int i = 0; i < arraylength; i++)
{
    arrayvalue[i] = i;
}
```

```
i = 0
if (i < arraylength)
 {
     arrayvalue[i] = i;
     i = i + 1;
     jump loop
}
else
     End
```

Trình biên dịch (3) - Quiz 1

- Biên dịch chương trình được viết bằng ngôn ngữ C sau sang hợp ngữ MIPS. Với arraylength và i tưng ứng với thanh ghi $s0 và $s1; mảng arrayvalue có địa chỉ base nằm trong thanh ghi $s5:

```
i = 0
if (i < arraylength)
 {
    arrayvalue[i] = i;
    i = i + 1;
    jump loop
}
else
    End
```

```
slt $t0, $s1, $s0
beq $t0, $zero, End
```

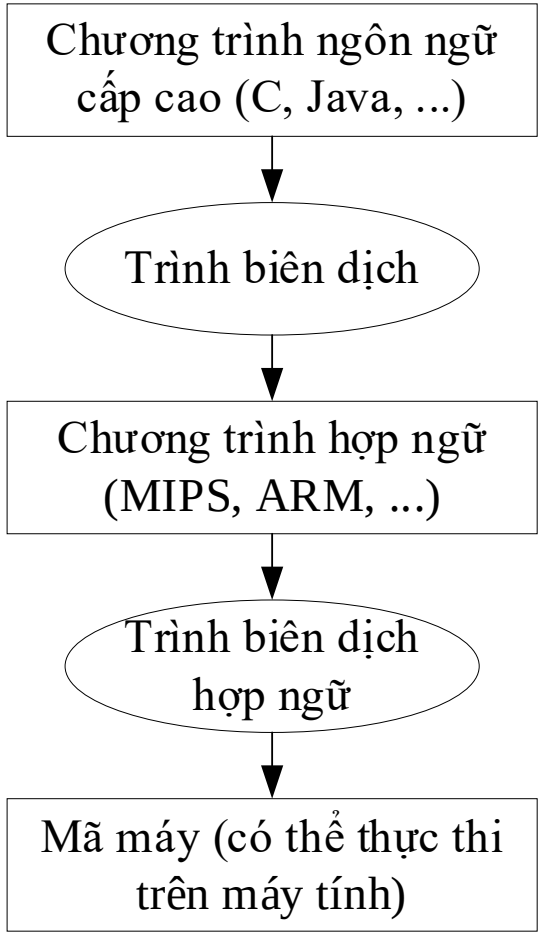
```
sll $t1, $s1, 2
add $t2, $t1, $s5
sw $s1, 0 ($t2)

M[0 + $t2] = $s1
```

```
sll $t1, $s1, 2
sw $s1, $t1 ($s5)

M[$t1+ $s5] = $s1
M[0 + $t1+ $s5] = $s1
```

```
sw $s1, $s1*4 ($s5)

M[$s1*4 + $s5] = $s1
```

Trình biên dịch hợp ngữ (trình thông dịch)

- **Trình biên dịch hợp ngữ** có chức năng chuyển chương trình được viết bởi hợp ngữ thành mã máy:
  - Mã máy là các chuỗi bit (0, 1) có thể được thực thi trên máy tính
- Có thể sử dụng lệnh giả (psudo instruction) để viết chương trình hợp ngữ nhằm đơn giản hơn cho lập trình viên
  - Lệnh giả: Không phải lệnh thực sự của máy nhưng trình biên dịch có thể chuyển thành lệnh thực sự



Chuyển đổi lệnh ASM MIPS sang mã máy

- Chuyển câu lệnh assembly (ASM) MIPS sau sang mã máy:

  and $t3, $s0, $s2

- Thực hiện việc chuyển đổi theo các bước sau:
  - Bước 1: Tra bảng "MIPS reference data" xem lệnh and thuộc định dạng nào => R type

And	and	R	R[rd] = R[rs] & R[rt]	$0 / 24_{hex}$

op	rs	rt	rd	shamt	funct

Chuyển đổi lệnh ASM MIPS sang mã máy (2)

- Chuyển câu lệnh assembly (ASM) MIPS sau sang mã máy:

  and $t3, $s0, $s2

- Thực hiện việc chuyển đổi theo các bước sau:
  - Bước 1: Tra bảng "MIPS reference data" xem lệnh and thuộc định dạng nào => R type
  - Bước 2: Tra các trường opcode và function

And	and	R	R[rd] = R[rs] & R[rt]	$0 / 24_{hex}$	

op	rs	rt	rd	shamt	funct
000000					100100

Chuyển đổi lệnh ASM MIPS sang mã máy (3)

- Chuyển câu lệnh assembly (ASM) MIPS sau sang mã máy:

  and $t3, $s0, $s2 => $t3 = $s0 & $s2

- Thực hiện việc chuyển đổi theo các bước sau:
  - Bước 1: Tra bảng "MIPS reference data" xem lệnh and thuộc định dạng nào => R type
  - Bước 2: Tra các trường opcode và function
  - Bước 3: Tra vị trí và chỉ số các thanh ghi

And	and	R R[rd] = R[rs] & R[rt]	$0 / 24_{hex}$

op	rs	rt	rd	shamt	funct
000000	10000	10010	01011		100100

NAME	NUMBER
$zero	0
$at	1
$v0-$v1	2-3
$a0-$a3	4-7
$t0-$t7	8-15
$s0-$s7	16-23
$t8-$t9	24-25
$k0-$k1	26-27
$gp	28
$sp	29
$fp	30
$ra	31

Chuyển đổi lệnh ASM MIPS sang mã máy (4)

- Chuyển câu lệnh assembly (ASM) MIPS sau sang mã máy:

  and $t3, $s0, $s2 => $t3 = $s0 & $s2

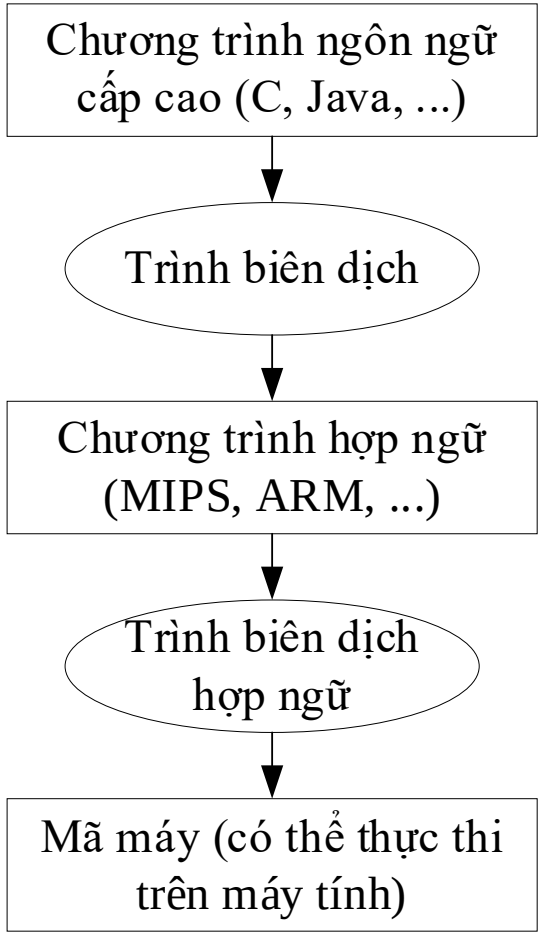
- Thực hiện việc chuyển đổi theo các bước sau:
  - Bước 1: Tra bảng "MIPS reference data" xem lệnh and thuộc định dạng nào => R type
  - Bước 2: Tra các trường opcode và function
  - Bước 3: Tra vị trí và chỉ số các thanh ghi
  - Bước 4: Điền trường shamt và hoàn thành mã máy của lệnh

op	rs	rt	rd	shamt	funct
000000	10000	10010	01011	00000	100100

Trình biên dịch hợp ngữ (2)

```
add  $t1, $t2, $t1
addi $t1, $a0, 0
bne  $a1, $t1, exit
lw   $a3, 4($t1)
exit:
```

```
0x01494820
0x20890000
0x14a90001
0x8d270004
...
```



Trình biên dịch hợp ngữ (2)

```
add  $t1, $t2, $t1
addi $t1, $a0, 0
bne  $a1, $t1, exit
lw   $a3, 4($t1)
exit:
```

```
0x01494820
0x20890000
0x14a90001
0x8d270004
...
```

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] + SignExtImm	(1,2)	8_{hex}
Add Imm. Unsigned	addiu	I	R[rt] = R[rs] + 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] & ZeroExtImm	(3)	c_{hex}
Branch On Equal	beq	I	if(R[rs]==R[rt]) PC=PC+4+BranchAddr	(4)	4_{hex}
Branch On Not Equal	bne	I	if(R[rs]!=R[rt]) PC=PC+4+BranchAddr	(4)	5_{hex}
Jump	j	J	PC=JumpAddr	(5)	2_{hex}
Jump And Link	jal	J	R[31]=PC+8;PC=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] +SignExtImm](7:0)}	(2)	24_{hex}
Load Halfword Unsigned	lhu	I	R[rt]={16'b0,M[R[rs] +SignExtImm](15:0)}	(2)	25_{hex}
Load Linked	ll	I	R[rt] = M[R[rs]+SignExtImm]	(2,7)	30_{hex}
Load Upper Imm.	lui	I	R[rt] = {imm, 16'b0}		f_{hex}
Load Word	lw	I	R[rt] = M[R[rs]+SignExtImm]	(2)	23_{hex}

Trình biên dịch hợp ngữ (3) - Quiz 2

- Biên dịch chương trình được viết bằng hợp ngữ MIPS bên cạnh sang mã máy, biết rằng chương trình bắt đầu ở địa chỉ 0x000C0:

```
bne   $s0, $s1, FAIL
add   $s2, $0, $0
j     END
FAIL: addi  $s2, $0, -1
END:
```

MIPS Reference Data

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] + SignExtImm	(1,2)	8_{hex}
Add Imm. Unsigned	addiu	I	R[rt] = R[rs] + 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] & ZeroExtImm	(3)	c_{hex}
Branch On Equal	beq	I	if(R[rs]==R[rt]) PC=PC+4+BranchAddr	(4)	4_{hex}
Branch On Not Equal	bne	I	if(R[rs]!=R[rt]) PC=PC+4+BranchAddr	(4)	5_{hex}
Jump	j	J	PC=JumpAddr	(5)	2_{hex}
Jump And Link	jal	J	R[31]=PC+8;PC=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] +SignExtImm](7:0)}	(2)	24_{hex}
Load Halfword Unsigned	lhu	I	R[rt]={16'b0,M[R[rs] +SignExtImm](15:0)}	(2)	25_{hex}
Load Linked	ll	I	R[rt] = M[R[rs]+SignExtImm]	(2,7)	30_{hex}
Load Upper Imm.	lui	I	R[rt] = {imm, 16'b0}		f_{hex}
Load Word	lw	I	R[rt] = M[R[rs]+SignExtImm]	(2)	23_{hex}

Biên dịch ngược

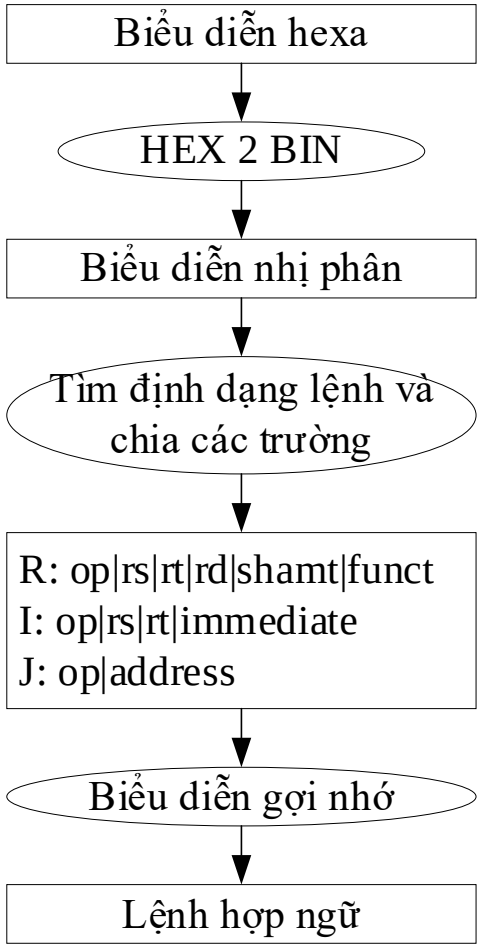
Biên dịch ngược là quá trình khôi phục mã máy thành chương trình hợp ngữ

0x00af8020

0000 0000 1010 1111 1000 0000 0010 0000

000000 00101 01111 10000 00000 100000

add $16, $5, $15 hoặc add $s0, $a1, $t7



Chuyển mã máy sang ngôn ngữ Assembly (ASM) MIPS

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] + SignExtImm	(1,2)	8_{hex}
Add Imm. Unsigned	addiu	I	R[rt] = R[rs] + 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}

- Chuyển mã máy sau sang câu lệnh assembly MIPS:

0x01304024

- Đáp án: Thực hiện việc chuyển đổi theo các bước sau:
  - Bước 1: Chuyển mã máy sang hệ nhị phân
  - Bước 2: Chọn 6bit đầu và tra opcode, nếu opcode bằng 000000 thì tra tiếp function 6bit cuối trong bảng MIPS

0x	0	1	3	0	4	0	2	4
	0000	0001	0011	0000	0100	0000	0010	0100

op		funct
000000	01 0011 0000 0100 0000 00	100100

Chuyển mã máy sang ngôn ngữ Assembly (ASM) MIPS

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] + SignExtImm	(1,2)	8_{hex}
Add Imm. Unsigned	addiu	I	R[rt] = R[rs] + 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}

- Chuyển mã máy sau sang câu lệnh assembly MIPS:

0x01304024

- Đáp án: Thực hiện việc chuyển đổi theo các bước sau:
  - Bước 2: Chọn 6bit đầu và tra opcode, nếu opcode bằng 000000 thì tra tiếp function 6bit cuối trong bảng MIPS
  - Bước 3: Tìm các thanh ghi tương ứng với các trường trong format R

op		funct
000000	01 0011 0000 0100 0000 00	100100

op	rs	rt	rd	shamt	funct
000000	01001	10000	01000	00000	100100
	9 ⇨ t1	16 ⇨ s0	8 ⇨ t0		

Chuyển mã máy sang ngôn ngữ Assembly (ASM) MIPS

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] + SignExtImm	(1,2)	8_{hex}
Add Imm. Unsigned	addiu	I	R[rt] = R[rs] + 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]		$0 / 24_{hex}$

- Chuyển mã máy sau sang câu lệnh assembly MIPS:

0x01304024

- Đáp án: Thực hiện việc chuyển đổi theo các bước sau:
  - Bước 2: Chọn 6bit đầu và tra opcode, nếu opcode bằng 000000 thì tra tiếp function 6bit cuối trong bảng MIPS
  - Bước 3: Tìm các thanh ghi tương ứng với các trường trong format R
  - Bước 4: Hoàn thành lệnh: and $t0, $t1, $s0

op	rs	rt	rd	shamt	funct
000000	01001	10000	01000	00000	100100
	9 ⇨ t1	16 ⇨ s0	8 ⇨ t0		

Bài tập

- Chuyển mã máy sau sang câu lệnh assembly MIPS:

  0x2128fff3

  0xad28fffc

Bài tập (2)

- Biên dịch chương trình chương trình được viết bằng ngôn ngữ lập trình C sau sang hợp ngữ MIPS, sau đó biên dịch sang mã máy với $s0 = count và địa chỉ base của arrayA và arrayB lần lượt là $s5 và $s6:

```
int count = 1;
while(count <= 20){
    arrayA[count - 1] = arrayB[count + 2];
    count++;
}
```

```
count = 1
if (count < 21)
 {
      arrayA[count - 1] = arrayB[count + 2];
       count = count + 1;
     jump if
}
End
```

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN – ĐHQG HCM

KHOA KỸ THUẬT MÁY TÍNH

TỔ CHỨC VÀ CẤU TRÚC MÁY TÍNH 2

ÔN TẬP
KIẾN TRÚC TẬP LỆNH

PHAN ĐÌNH DUY

TP. Hồ Chí Minh, ngày 05 tháng 9 năm 2022

FACULTY OF COMPUTER ENGINEERING

NỘI DUNG

- Thực thi chương trình
- Chuyển từ C sang MIPS
- Chuyển từ MIPS sang C
- Chuyển từ MIPS sang mã máy
- Chuyển từ mã máy sang MIPS
- Bài tập

Thực thi chương trình

- Giá trị của thanh ghi $v0 và $v1 là bao nhiêu sau khi thực thi chương trình bên dưới

```
lui  $t0, 0x5678
addi $t1, $0, 0x4321
or   $a0, $t0, $t1
nor  $a1, $a0, $0
slt  $v0, $t0, $a0
sltu $v1, $t1, $a1
```

Thực thi chương trình (2)

- Giá trị của thanh ghi $v0 và $v1 là bao nhiêu sau khi thực thi chương trình bên dưới

```
addi $a0, $0, 0x1234
addi $a1, $0, 0xCAFE
addi $t1, $0, 0x432C
sw   $a0, 0($t1)
addi $t1, $t1, 4
sw   $a1, 4($t1)
lw   $v0, -8($t1)
lw   $v1, 0($t1)
```

Chuyển từ C sang MIPS

- Cho chương trình C bên dưới. Biết rằng g, h, i, j là những biến nguyên 32 bit, trả lời các câu hỏi sau:
  - Tìm mã hợp ngữ MIPS tương đương của chương trình.
  - Cần bao nhiêu lệnh MIPS để hiện thực chương trình
  - Nếu các biến g, h, i và j có giá trị tương ứng là 1, 2, 3, 4, 5 thì giá trị của f và k là bao nhiêu?

$$f = g + h + i + j$$
$$k = g + (h + 5)$$

Chuyển từ C sang MIPS (2)

Cho câu lệnh C: f = g – A[B[4]];

- Tìm mã MIPS tương đương của chương trình nếu địa chỉ của mảng A và B lần lượt nằm trong các thanh ghi $s6 và $s7. Biến g là biến nguyên 32 bit.

Chuyển từ MIPS sang C

- Tìm chương trình C tương ứng với chương trình hợp ngữ MIPS bên dưới:

```
add  $t0, $a0, $a1
addi $t1, $a0, 5
sub  $t2, $t0, $a1
add  $s0, $t2, $a2
```

Chuyển từ MIPS sang C (2)

- Tìm chương trình C tương ứng với chương trình hợp ngữ MIPS bên dưới:

```
        bne  $a0, $a1, another
        add  $s0, $0, $0
        j exit
another: addi  $s0, $s0, -1
        exit:
```

Chuyển từ MIPS sang mã máy

- Chuyển chương trình hợp ngữ MIPS bên dưới sang mã máy

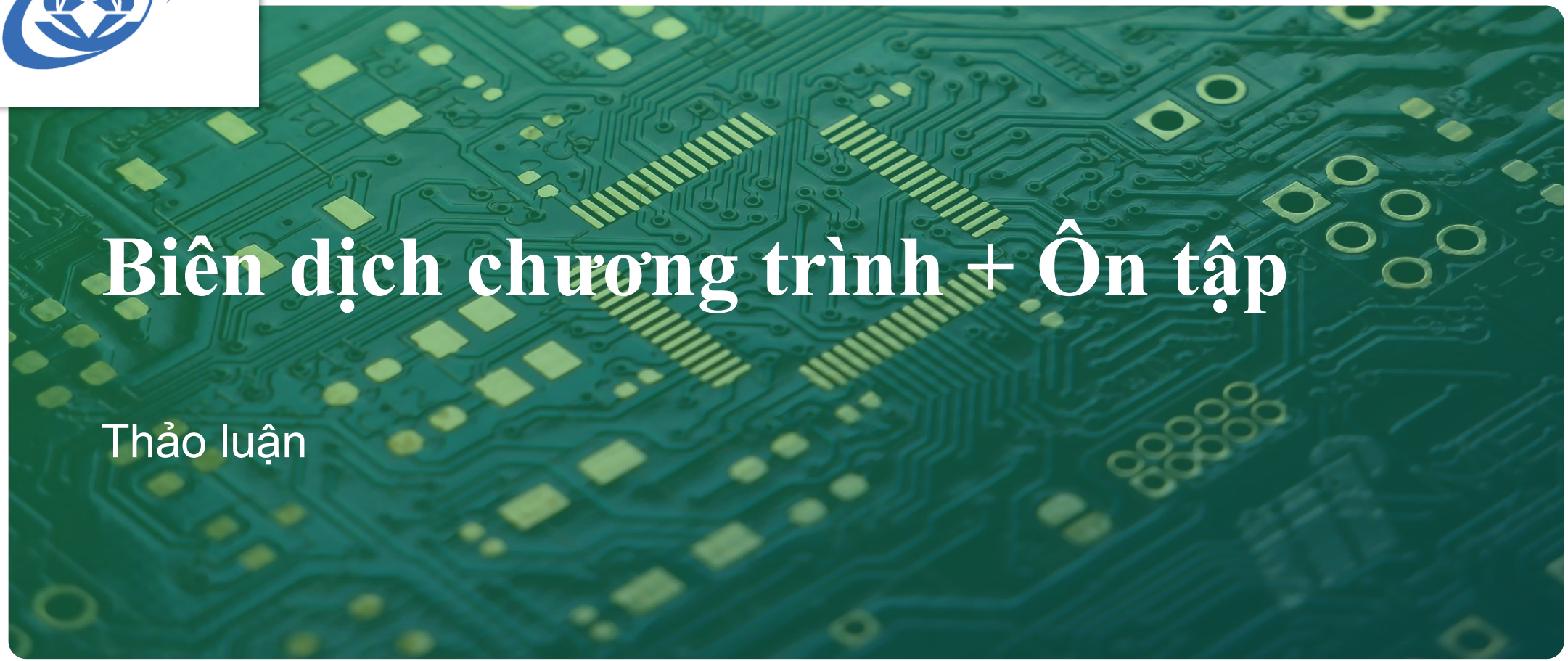
```
        add $t0, $t0, $zero
Loop:   lw $t1, 4($s3)
        addi    $s3, $s3, 4
        bne     $t1, $t0, Loop
```

Chuyển từ mã máy sang MIPS

- Chuyển chương trình được lưu trong bộ nhớ bên dưới sang hợp ngữ MIPS

0x00a6202a

0x2149ff90



Biên dịch chương trình + Ôn tập

Thảo luận