

# Assemblaggio di un programma MIPS

---

Passate

Esempio

# Passate

---

- **Il processo di assemblaggio procede secondo diversi passaggi logici:**
- **Espansione / sostituzione**
  - Ogni pseudo-istruzione viene espansa in una sequenza di istruzioni native
  - I registri simbolici vengono sostituiti con i registri numerici
- **Allocazione dati e istruzioni**
  - Allocazione dei dati specificati nella sezione .data
  - Allocazione delle istruzioni nella sezione .text
- **Costruzione della tabella dei simboli**
  - Ad ogni simbolo viene associato un indirizzo
- **Assemblaggio**
  - Le istruzioni vengono assemblate
- **Creazione file binario**
  - Il risultato dell'assemblaggio viene strutturato secondo lo specifico formato di uscita
  - Per esempio il formato elf, hex, axf, ...

# Programma

- Consideriamo il seguente programma

- Somma i primi sei valori della sequenza di Fibonacci presenti nel vettore V
- Salva il risultato nella variabile SUM

```
.data
V:      .byte   1, 1, 2, 3, 5, 8
        .align  2
SUM:    .word   5

.text
MAIN:   la $t0, V
        li $t1, 6
        li $t2, 0
LOOP:   lb $t3, ($t0)
        add     $t2, $t2, $t3
        addi    $t0, $t0, +1
        addi    $t1, $t1, -1
        bne     $t1, $zero, LOOP
        la $t0, SUM
        sw $t2, ($t0)
```

# Passo 1: Espansione pseudo-istruzioni

- Vengono espanso le pseudo-istruzioni
- I nomi simbolici dei registri sono mantenuti
- Le scritture simboliche high(X) e low(X) indicano le half-word alta e bassa del valore del simbolo X

```
.data
V:      .byte 1, 1, 2, 3, 5, 8
        .align 2
SUM:    .word 5

        .text
MAIN:   la    $t0, V           # 1
        li    $t1, 6          # 2
        li    $t2, 0          # 3
LOOP:  lb    $t3, ($t0)       # 4
        add   $t2, $t2, $t3   # 5
        addi  $t0, $t0, +1    # 6
        addi  $t1, $t1, -1    # 7
        bne  $t1, $zero, LOOP # 8
        la    $t0, SUM        # 9
        sw   $t2, ($t0)       # 10
```

```
.data
V:      .byte 1, 1, 2, 3, 5, 8
        .align 2
SUM:    .word 5

        .text
MAIN:   lui   $at, high(V)    # 1
        ori   $t0, $at, low(V) # 1
        addiu $t1, $zero, 6   # 2
        addiu $t2, $zero, 0   # 3
LOOP:  lb    $t3, ($t0)       # 4
        add   $t2, $t2, $t3   # 5
        addi  $t0, $t0, +1    # 6
        addi  $t1, $t1, -1    # 7
        bne  $t1, $zero, LOOP # 8
        lui   $at, high(SUM)  # 9
        ori   $t0, $at, low(SUM) # 9
        sw   $t2, ($t0)       # 10
```

# Passo 2: Sostituzione dei registri

- I nomi simbolici dei registri sono sostituiti con i valori numerici

```
.data
V:      .byte 1, 1, 2, 3, 5, 8
       .align 2
SUM:    .word 5

.text
MAIN:   lui    $at, high(V)           # 1
       ori    $t0, $at, low(V)       # 1
       addiu  $t1, $zero, 6          # 2
       addiu  $t2, $zero, 0          # 3
LOOP:   lb     $t3, ($t0)             # 4
       add    $t2, $t2, $t3          # 5
       addi   $t0, $t0, +1           # 6
       addi   $t1, $t1, -1           # 7
       bne   $t1, $zero, LOOP        # 8
       lui   $at, high(SUM)         # 9
       ori   $t0, $at, low(SUM)     # 9
       sw    $t2, ($t0)             # 10
```

```
.data
V:      .byte 1, 1, 2, 3, 5, 8
       .align 2
SUM:    .word 5

.text
MAIN:   lui    $1, high(V)           # 1
       ori    $8, $1, low(V)        # 1
       addiu  $9, $0, 6              # 2
       addiu  $10, $0, 0             # 3
LOOP:   lb     $11, ($8)             # 4
       add    $10, $10, $11          # 5
       addi   $8, $8, +1             # 6
       addi   $9, $9, -1             # 7
       bne   $9, $0, LOOP            # 8
       lui   $1, high(SUM)          # 9
       ori   $8, $1, low(SUM)       # 9
       sw    $10, ($8)              # 10
```

# Passo 3: Allocazione dati

- La sezione `.data` inizia all'indirizzo `0x10010000`

```
.data
V:    .byte 1, 1, 2, 3, 5, 8
      .align 2
SUM:  .word 5

      .text
MAIN: lui    $1, high(V)           # 1
      ori    $8, $1, low(V)        # 1
      addiu  $9, $0, 6             # 2
      addiu  $10, $0, 0            # 3
LOOP: lb     $11, ($8)             # 4
      add    $10, $10, $11         # 5
      addi   $8, $8, +1            # 6
      addi   $9, $9, -1           # 7
      bne   $9, $0, LOOP          # 8
      lui    $1, high(SUM)        # 9
      ori    $8, $1, low(SUM)     # 9
      sw    $10, ($8)             # 10
```

0x10010000		

# Passo 3: Allocazione dati

- Alloca e inizializza 6 byte del vettore V

```
.data
V:    .byte 1, 1, 2, 3, 5, 8
      .align 2
SUM:  .word 5

.text
MAIN: lui    $1,  high(V)           # 1
      ori    $8,  $1,  low(V)      # 1
      addiu  $9,  $0,  6           # 2
      addiu  $10, $0,  0           # 3
LOOP: lb     $11, ($8)             # 4
      add   $10, $10, $11         # 5
      addi  $8,  $8,  +1          # 6
      addi  $9,  $9,  -1          # 7
      bne  $9,  $0,  LOOP         # 8
      lui  $1,  high(SUM)        # 9
      ori  $8,  $1,  low(SUM)    # 9
      sw   $10, ($8)            # 10
```

0x10010000	V	0x03020101
0x10010004		0x00000805

# Passo 3: Allocazione dati

- Allinea l'indirizzo alla prossima parola

```
.data
V:      .byte  1, 1, 2, 3, 5, 8
        .align 2
SUM:    .word  5

        .text
MAIN:   lui    $1,  high(V)           # 1
        ori    $8,  $1,  low(V)      # 1
        addiu  $9,  $0,  6           # 2
        addiu  $10, $0,  0           # 3
LOOP:   lb     $11, ($8)             # 4
        add    $10, $10, $11        # 5
        addi   $8,  $8,  +1          # 6
        addi   $9,  $9,  -1          # 7
        bne   $9,  $0,  LOOP         # 8
        lui    $1,  high(SUM)       # 9
        ori    $8,  $1,  low(SUM)   # 9
        sw     $10, ($8)            # 10
```

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008		



# Passo 3: Allocazione dati

- Alloca ed inizializza la parola SUM

```
.data
V:      .byte  1, 1, 2, 3, 5, 8
        .align 2
SUM:    .word  5

.text
MAIN:   lui    $1,  high(V)          # 1
        ori    $8,  $1,  low(V)     # 1
        addiu  $9,  $0,  6          # 2
        addiu  $10, $0,  0          # 3
LOOP:   lb     $11, ($8)            # 4
        add    $10, $10, $11        # 5
        addi   $8,  $8,  +1         # 6
        addi   $9,  $9,  -1         # 7
        bne   $9,  $0,  LOOP        # 8
        lui    $1,  high(SUM)       # 9
        ori    $8,  $1,  low(SUM)   # 9
        sw     $10, ($8)            # 10
```

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000

# Passo 4: Allocazione istruzioni

- La sezione `.text` inizia all'indirizzo `0x00400000`

```
.data
V:      .byte  1, 1, 2, 3, 5, 8
        .align 2
SUM:    .word  5

        .text
MAIN:   lui    $1,  high(V)           # 1
        ori   $8,  $1,  low(V)      # 1
        addiu $9,  $0,  6            # 2
        addiu $10, $0,  0           # 3
LOOP:   lb    $11, ($8)             # 4
        add   $10, $10, $11         # 5
        addi  $8,  $8,  +1          # 6
        addi  $9,  $9,  -1          # 7
        bne  $9,  $0,  LOOP         # 8
        lui   $1,  high(SUM)        # 9
        ori   $8,  $1,  low(SUM)    # 9
        sw   $10, ($8)              # 10
```

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000		

# Passo 4: Allocazione istruzioni

- Le istruzioni e le relative etichette vengono ordinatamente allocate in memoria

```
.data
V:      .byte  1, 1, 2, 3, 5, 8
        .align 2
SUM:    .word  5

        .text
MAIN:   lui    $1,  high(V)           # 1
        ori    $8,  $1, low(V)       # 1
        addiu  $9,  $0, 6            # 2
        addiu  $10, $0, 0            # 3
LOOP:   lb     $11, ($8)             # 4
        add    $10, $10, $11        # 5
        addi   $8,  $8, +1           # 6
        addi   $9,  $9, -1          # 7
        bne   $9,  $0, LOOP         # 8
        lui    $1,  high(SUM)       # 9
        ori    $8,  $1, low(SUM)    # 9
        sw    $10, ($8)             # 10
```

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000	MAIN	lui    \$1, high(V)
0x00400004		ori    \$8, \$1, low(V)
0x00400008		addiu  \$9, \$0, 6
0x0040000C		addiu  \$10, \$0, 0
0x00400010	LOOP	lb     \$11, (\$8)
0x00400014		add    \$10, \$10, \$11
0x00400018		addi   \$8, \$8 +1
0x0040001C		addi   \$9, \$9, -1
0x00400020		bne   \$9, \$0, LOOP
0x00400024		lui    \$1, high(SUM)
0x00400028		ori    \$8, \$1, low(SUM)
0x0040002c		sw    \$10, (\$8)

# Passo 5: Costruzione tabella dei simboli

- Si costruisce la tabella dei simboli completa

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000	MAIN	lui \$1, high(V)
0x00400004		ori \$8, \$1, low(V)
0x00400008		addiu \$9, \$0, 6
0x0040000C		addiu \$10, \$0, 0
0x00400010	LOOP	lb \$11, (\$8)
0x00400014		add \$10, \$10, \$11
0x00400018		addi \$8, \$8 +1
0x0040001C		addi \$9, \$9, -1
0x00400020		bne \$9, \$0, LOOP
0x00400024		lui \$1, high(SUM)
0x00400028		ori \$8, \$1, low(SUM)
0x0040002c		sw \$10, (\$8)

Symbol	Address
V	0x10010000
SUM	0x10010008
MAIN	0x00400000
LOOP	0x00400010

# Passo 6: Traduzione

## Istruzione: lui

- Tipo: I: Immediate
- Opcode: 001111
- Registri: rs = N/A, 00000      rt = 1, 00001
- Immediato: V = 0x10010000      high(V) = 0x1001

opcode			rs				rt				imm																						
0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3			c				0				1				1				0				0				1						

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000	MAIN	lui \$1, high(V)
0x00400004		ori \$8, \$1, low(V)
0x00400008		addiu \$9, \$0, 6
0x0040000C		addiu \$10, \$0, 0
0x00400010	LOOP	lb \$11, (\$8)
0x00400014		add \$10, \$10, \$11
0x00400018		addi \$8, \$8 +1
0x0040001C		addi \$9, \$9, -1
0x00400020		bne \$9, \$0, LOOP
0x00400024		lui \$1, high(SUM)
0x00400028		ori \$8, \$1, low(SUM)
0x0040002c		sw \$10, (\$8)

Symbol	Address
V	0x10010000
SUM	0x10010008
MAIN	0x00400000
LOOP	0x00400010

# Passo 6: Traduzione

## ■ Istruzione: ori

- Tipo: I: Immediate
- Opcode: 001101
- Registri: rs = 1, 00001      rt = 8, 01000
- Immediato: V = 0x10010000      low(V) = 0x0000

opcode						rs				rt				imm																					
0	0	1	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3						4				2				8				0				0				0				0					

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000	MAIN	0x3C011001
0x00400004		ori \$8, \$1, low(V)
0x00400008		addiu \$9, \$0, 6
0x0040000C		addiu \$10, \$0, 0
0x00400010	LOOP	lb \$11, (\$8)
0x00400014		add \$10, \$10, \$11
0x00400018		addi \$8, \$8 +1
0x0040001C		addi \$9, \$9, -1
0x00400020		bne \$9, \$0, LOOP
0x00400024		lui \$1, high(SUM)
0x00400028		ori \$8, \$1, low(SUM)
0x0040002c		sw \$10, (\$8)

Symbol	Address
V	0x10010000
SUM	0x10010008
MAIN	0x00400000
LOOP	0x00400010

# Passo 6: Traduzione

## Istruzione: addiu

- Tipo: I: Immediate
- Opcode: 001001
- Registri: rs = 0, 00000      rt = 9, 01001
- Immediato: 6 = 0x00000006

opcode						rs				rt				imm																				
0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
2						4				0				9				0				0				0				6				

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000	MAIN	0x3C011001
0x00400004		0x34280000
0x00400008		addiu \$9, \$0, 6
0x0040000C		addiu \$10, \$0, 0
0x00400010	LOOP	lb \$11, (\$8)
0x00400014		add \$10, \$10, \$11
0x00400018		addi \$8, \$8 +1
0x0040001C		addi \$9, \$9, -1
0x00400020		bne \$9, \$0, LOOP
0x00400024		lui \$1, high(SUM)
0x00400028		ori \$8, \$1, low(SUM)
0x0040002c		sw \$10, (\$8)

Symbol	Address
V	0x10010000
SUM	0x10010008
MAIN	0x00400000
LOOP	0x00400010

# Passo 6: Traduzione

## ■ Istruzione: addiu

- Tipo: I: Immediate
- Opcode: 001001
- Registri: rs = 0, 00000      rt = 10, 01010
- Immediato: 0 = 0x00000000

opcode						rs				rt				imm																			
0	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2						4				0				A				0				0				0				0			

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000	MAIN	0x3C011001
0x00400004		0x34280000
0x00400008		0x24090006
0x0040000c		addiu \$10, \$0, 0
0x00400010	LOOP	lb \$11, (\$8)
0x00400014		add \$10, \$10, \$11
0x00400018		addi \$8, \$8 +1
0x0040001c		addi \$9, \$9, -1
0x00400020		bne \$9, \$0, LOOP
0x00400024		lui \$1, high(SUM)
0x00400028		ori \$8, \$1, low(SUM)
0x0040002c		sw \$10, (\$8)

Symbol	Address
V	0x10010000
SUM	0x10010008
MAIN	0x00400000
LOOP	0x00400010



# Passo 6: Traduzione

## ▪ Istruzione: lb

- Tipo: I: Immediate
- Opcode: 001001
- Registri: rs = 8, 01000                      rt = 11, 01011
- Immediato: 0 = 0x00000000

opcode						rs				rt				imm																						
1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8						1				0				B				0				0				0				0						

0x10010000	V		0x03020101
0x10010004			0x00000805
0x10010008	SUM		0x00000000
0x00400000	MAIN		0x3C011001
0x00400004			0x34280000
0x00400008			0x24090006
0x0040000C			0x240A0000
0x00400010	LOOP	lb	\$11, (\$8)
0x00400014		add	\$10, \$10, \$11
0x00400018		addi	\$8, \$8 +1
0x0040001C		addi	\$9, \$9, -1
0x00400020		bne	\$9, \$0, LOOP
0x00400024		lui	\$1, high(SUM)
0x00400028		ori	\$8, \$1, low(SUM)
0x0040002c		sw	\$10, (\$8)

Symbol	Address
V	0x10010000
SUM	0x10010008
MAIN	0x00400000
LOOP	0x00400010

# Passo 6: Traduzione

## ▪ Istruzione: add

- Tipo: R: Register
- Opcode: 000000      funct = 100000      shamt = 0000
- Registri: rs = 10, 01010      rt = 11, 01011      rd = 10, 01010
- Immediato: 0 = 0x00000000

opcode						rs				rt				rd					shamt					funct											
0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0			
0						1				4				5					0					2						0					

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000	MAIN	0x3C011001
0x00400004		0x34280000
0x00400008		0x24090006
0x0040000C		0x240A0000
0x00400010	LOOP	0x810B0000
0x00400014		add \$10, \$10, \$11
0x00400018		addi \$8, \$8, +1
0x0040001C		addi \$9, \$9, -1
0x00400020		bne \$9, \$0, LOOP
0x00400024		lui \$1, high(SUM)
0x00400028		ori \$8, \$1, low(SUM)
0x0040002c		sw \$10, (\$8)

Symbol	Address
V	0x10010000
SUM	0x10010008
MAIN	0x00400000
LOOP	0x00400010

# Passo 6: Traduzione

## ▪ Istruzione: addi

- Tipo: I: Immediate
- Opcode: 001000
- Registri: rs = 8, 01000      rt = 8, 01000
- Immediato: 1 = 0x00000001

opcode						rs				rt				imm																			
0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2		1		0		8		0		0		0		1																			

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000	MAIN	0x3C011001
0x00400004		0x34280000
0x00400008		0x24090006
0x0040000C		0x240A0000
0x00400010	LOOP	0x810B0000
0x00400014		0x014B5020
0x00400018		addi \$8, \$8 +1
0x0040001C		addi \$9, \$9, -1
0x00400020		bne \$9, \$0, LOOP
0x00400024		lui \$1, high(SUM)
0x00400028		ori \$8, \$1, low(SUM)
0x0040002c		sw \$10, (\$8)

Symbol	Address
V	0x10010000
SUM	0x10010008
MAIN	0x00400000
LOOP	0x00400010

# Passo 6: Traduzione

## ■ Istruzione: addi

- Tipo: I: Immediate
- Opcode: 001000
- Registri: rs = 9, 01001      rt = 9, 01001
- Immediato: -1 = 0xFFFFFFFF

opcode						rs				rt				imm																						
0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2		1	2		9			F			F			F			F																			

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000	MAIN	0x3C011001
0x00400004		0x34280000
0x00400008		0x24090006
0x0040000C		0x240A0000
0x00400010	LOOP	0x810B0000
0x00400014		0x014B5020
0x00400018		0x21080001
0x0040001C		addi \$9, \$9, -1
0x00400020		bne \$9, \$0, LOOP
0x00400024		lui \$1, high(SUM)
0x00400028		ori \$8, \$1, low(SUM)
0x0040002c		sw \$10, (\$8)

Symbol	Address
V	0x10010000
SUM	0x10010008
MAIN	0x00400000
LOOP	0x00400010

# Passo 6: Traduzione

## Istruzione: bne

- Tipo: I: Immediate
- Opcode: 000101
- Registri: rs = 9, 01001      rt = 0, 00000
- Immediato: (LOOP - PC - 4) >> 2 = (0x00400010 - 0x00400020 - 4) >> 2 = 0xFFFB

opcode						rs					rt					imm																
0	0	0	1	0	1	0	1	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	
1						5					2					0					F			F			F			B		

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000	MAIN	0x3C011001
0x00400004		0x34280000
0x00400008		0x24090006
0x0040000C		0x240A0000
0x00400010	LOOP	0x810B0000
0x00400014		0x014B5020
0x00400018		0x21080001
0x0040001C		0x2129FFFF
0x00400020		bne \$9, \$0, LOOP
0x00400024		lui \$1, high(SUM)
0x00400028		ori \$8, \$1, low(SUM)
0x0040002c		sw \$10, (\$8)

Symbol	Address
V	0x10010000
SUM	0x10010008
MAIN	0x00400000
LOOP	0x00400010

# Passo 6: Traduzione

## ■ Istruzione: lui

- Tipo: I: Immediate
- Opcode: 001111
- Registri: rs = N/A, 00000      rt = 1, 00001
- Immediato: SUM=0x10010008    high(SUM) = 0x1001

opcode			rs				rt				imm																							
0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3			c				0				1				1				0				0				1							

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000	MAIN	0x3C011001
0x00400004		0x34280000
0x00400008		0x24090006
0x0040000C		0x240A0000
0x00400010	LOOP	0x810B0000
0x00400014		0x014B5020
0x00400018		0x21080001
0x0040001C		0x2129FFFF
0x00400020		0x1520FFFF
0x00400024		lui    \$1, high(SUM)
0x00400028		ori    \$8, \$1, low(SUM)
0x0040002c		sw    \$10, (\$8)

Symbol	Address
V	0x10010000
SUM	0x10010008
MAIN	0x00400000
LOOP	0x00400010

# Passo 6: Traduzione

## ■ Istruzione: ori

- Tipo: I: Immediate
- Opcode: 001101
- Registri: rs = 1, 00001      rt = 8, 01000
- Immediato: SUM=0x10010008    low(SUM) = 0x0008

opcode						rs				rt				imm																			
0	0	1	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
3						4				2				8				0				0				0				8			

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000	MAIN	0x3C011001
0x00400004		0x34280000
0x00400008		0x24090006
0x0040000C		0x240A0000
0x00400010	LOOP	0x810B0000
0x00400014		0x014B5020
0x00400018		0x21080001
0x0040001C		0x2129FFFF
0x00400020		0x1520FFFF
0x00400024		0x3c011001
0x00400028		ori \$8, \$1, low(SUM)
0x0040002c		sw \$10, (\$8)

Symbol	Address
V	0x10010000
SUM	0x10010008
MAIN	0x00400000
LOOP	0x00400010

# Passo 6: Traduzione

## ■ Istruzione: sw

- Tipo: I: Immediate
- Opcode: 101011
- Registri: rs = 8, 01000      rt = 10, 01010
- Immediato: 0 = 0x00000000

opcode					rs				rt				imm																							
1	0	1	0	1	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A					D				0				A				0				0				0				0							

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000	MAIN	0x3C011001
0x00400004		0x34280000
0x00400008		0x24090006
0x0040000C		0x240A0000
0x00400010	LOOP	0x810B0000
0x00400014		0x014B5020
0x00400018		0x21080001
0x0040001C		0x2129FFFF
0x00400020		0x1520FFFF
0x00400024		0x3c011001
0x00400028		0x34280008
0x0040002c		sw \$10, (\$8)

Symbol	Address
V	0x10010000
SUM	0x10010008
MAIN	0x00400000
LOOP	0x00400010



# Passo 6: Traduzione

---

## Finalizzazione

- La mappa binaria della memoria viene infine strutturata secondo il formato di uscita

0x10010000	V	0x03020101
0x10010004		0x00000805
0x10010008	SUM	0x00000000
0x00400000	MAIN	0x3C011001
0x00400004		0x34280000
0x00400008		0x24090006
0x0040000C		0x240A0000
0x00400010	LOOP	0x810B0000
0x00400014		0x014B5020
0x00400018		0x21080001
0x0040001C		0x2129FFFF
0x00400020		0x1520FFFF
0x00400024		0x3c011001
0x00400028		0x34280008
0x0040002c		0xAD0A0000