

1) processo di assemblaggio

Dati i due moduli assembler seguenti, **si compilino** le tabelle relative a:

1. i due moduli oggetto MAIN e MODULE
2. le basi di rilocalizzazione del codice e dei dati di entrambi i moduli
2. la tabella globale dei simboli
4. la tabella di impostazione del calcolo delle costanti e degli spiazamenti di istruzione e di dato
5. la tabella del codice eseguibile

modulo MAIN	modulo MODULE
<pre> .data STRUCT: .space 20 VECT: .space 12 INT: .word 23 .text .globl MAIN MAIN: li t0, 0xFFFF0ABCC la t2, STRUCT sw t0, 0(t2) la t3, VECT lw t1, 0(t3) beq t0, t0, MODULE MAINEND: j MAIN </pre>	<pre> .data ALPHA: .byte 'Y' # codice ASCII 'Y' = dec 89 .text .globl MODULE RESTART: la t4, INT lw t4, 0(t4) MODULE: la t5, ALPHA lb t6, 0(t5) sub t6, t6, t4 beq t6, zero, RESTART MODEND: j MAINEND </pre>

Regola generale per la compilazione di **tutte** le tabelle contenenti codice:

- espandere tutte le pseudo istruzioni
- l'allocazione delle variabili in memoria non è allineata (non c'è frammentazione di memoria)
- i codici operativi e i nomi dei registri vanno indicati in formato simbolico tutte le costanti numeriche all'interno del codice vanno indicate in esadecimale, con o senza prefisso 0x, e di lunghezza giusta per il codice che rappresentano
esempio: un'istruzione come `addi t0, t0, 15` è rappresentata: `addi t0, t0, 0x 00F`
- nei moduli oggetto i valori numerici che non possono essere indicati poiché dipendono dalla rilocalizzazione successiva, vanno posti a zero e avranno un valore definitivo nel codice eseguibile

(1) – moduli oggetto

modulo MAIN		
dimensione testo: 28 hex (40 dec)		
dimensione dati: 24 hex (36 dec)		
testo		
indirizzo	istruzione	
0	lui t0, 0x FFF0 B	
4	addi t0, 0xBCC	
8	auipc t2, 0x 0 0000	
C	addi t2, 0x000	
10	sw t0, 0(t2)	
14	auipc t3, 0x 0 0000	
18	addi t3, 0x000	
1C	lw t1, 0(t3)	
20	beq t0, t0, 0x 000	
24	jal zero, 0x0 0000	
dati		
indirizzo	contenuto (in esadecimale)	
0	non specificato	
14	non specificato	
20	0x 0000 0017 (valore su 32 bit) (23 dec)	
tabella dei simboli		
tipo può essere <i>T</i> (testo) oppure <i>D</i> (dato)		
simbolo	tipo	Valore
STRUCT	D	0x 0000 0000 0000 0000
VECT	D	0x 0000 0000 0000 0014
INT	D	0x 0000 0000 0000 0020
MAIN	T	0x 0000 0000 0000 0000
MAINEND	T	0x 0000 0000 0000 0024

modulo MODULE		
dimensione testo: 24 hex (36 dec)		
dimensione dati: 1 hex (1 dec)		
testo		
indirizzo	istruzione	
0	auipc t4, 0x 0000 0	
4	addi t4, 0x000	
8	lw t4, 0(t4)	
C	auipc t5, 0x 0000 0	
10	addi t5, 0x000	
14	lb t6, 0(t5)	
18	sub t6, t6, t4	
1C	beq t6, zero, 0x FF2	
20	jal zero, 0x 0000 0	
dati		
indirizzo	contenuto (in esadecimale)	
0	0x59 (valore su 8 bit) (89 dec)	
tabella dei simboli		
tipo può essere <i>T</i> (testo) oppure <i>D</i> (dato)		
simbolo	tipo	Valore
ALPHA	D	0x 0000 0000 0000 0000
RESTART	T	0x 0000 0000 0000 0000
MODULE	T	0x 0000 0000 0000 000C
MODEND	T	0x 0000 0000 0000 0020

tabella di rilocalizzazione			tabella di rilocalizzazione		
indirizzo	cod. operativo	simbolo	indirizzo	cod. operativo	simbolo
8	auipc	STRUCT	0	auipc	INT
C	addi	STRUCT	4	addi	INT
14	auipc	VECT	C	auipc	ALPHA
18	addi	VECT	10	addi	ALPHA
20	beq	MODULE	20	jal	MAINEND
24	jal	MAIN			

(2) – posizione in memoria dei moduli

<i>MAIN</i>	<i>MODULE</i>
base del testo: 0x 0000 0000 0040 0000	base del testo: 0x 0000 0000 0040 0028
base dei dati: 0x 0000 0000 1000 0000	base dei dati: 0x 0000 0000 1000 0024

(3) – tabella globale dei simboli

simbolo	valore iniziale	base	valore finale
STRUCT	0x 0000 0000 0000 0000	0x 0000 0000 1000 0000	0x 0000 0000 1000 0000
VECT	0x 0000 0000 0000 0014	0x 0000 0000 1000 0000	0x 0000 0000 1000 0014
INT	0x 0000 0000 0000 0020	0x 0000 0000 1000 0000	0x 0000 0000 1000 0020
MAIN	0x 0000 0000 0000 0000	0x 0000 0000 0040 0000	0x 0000 0000 0040 0000
MAINEND	0x 0000 0000 0000 0024	0x 0000 0000 0040 0000	0x 0000 0000 0040 0024
ALPHA	0x 0000 0000 0000 0000	0x 0000 0000 1000 0024	0x 0000 0000 1000 0024
RESTART	0x 0000 0000 0000 0000	0x 0000 0000 0040 0028	0x 0000 0000 0040 0028
MODULE	0x 0000 0000 0000 000C	0x 0000 0000 0040 0028	0x 0000 0000 0040 0034
MODEND	0x 0000 0000 0000 0020	0x 0000 0000 0040 0028	0x 0000 0000 0040 0048

(4) impostazione calcolo delle costanti e degli spiazziamenti di istruzione e di dato

modulo MAIN	modulo MODULE
lui #hi(FFF0ABCC) = 0xFFFF0A + 1 (20 bit sup + 1 (offset negativo, 11 bit = 1) = 0xFFFF0 B (20 bit)	auipc #pcrel_hi(INT) = 0x 0000 0000 1000 0020 - 0x 0000 0000 0040 0028 = 0x 0000 0000 0FBF FFF8 = 0x0FC0 0 (20 bit) (offset 11 = 1 -> offset negativo)
addi #lo(FFF0ABCC) = 0X BCC (12 bit)	addi #pcrel_lo(INT) = 0x 0000 0000 1000 0020 - 0x 0000 0000 0040 0028 = 0x 0000 0000 0FBF FFF8 = 0xFF8 (12 bit inferiori)
auipc #pcrel_hi(STRUCT) = 0x 0000 0000 1000 0000 - 0x 0000 0000 0040 0008 = 0x 0000 0000 0FBF FFF8 = 0x0FC0 0 (20 bit) (offset 11 = 1 -> offset negativo)	auipc #pcrel_hi(ALPHA) = 0x 0000 0000 1000 0024 - 0x 0000 0000 0040 0034 = 0x 0000 0000 0FBF FFF0 = 0x0FC0 0 (20 bit) (offset 11 = 1 -> offset negativo)
addi #pcrel_lo(STRUCT) = 0x 0000 0000 1000 0000 - 0x 0000 0000 0040 0008 = 0x 0000 0000 0FBF FFF8 = 0xFF8 (12 bit inferiori)	addi #pcrel_lo(ALPHA) = 0x 0000 0000 1000 0024 - 0x 0000 0000 0040 0034 = 0x 0000 0000 0FBF FFF0 = 0xFF0 (12 bit inferiori)
auipc #pcrel_hi(VECT) = 0x 0000 0000 1000 0014 - 0x 0000 0000 0040 0014 = 0x 0000 0000 0FC0 0000 = 0x0FC0 0 (20 bit sup)	beq #pcrel(RESTART)/2= (0x 0000 0000 0040 0028 - 0x 0000 0000 0040 0044)/2 = (0x 0000 0000 0000 0FE4)/2 = 0xFF2 (12 bit)
addi #pcrel_lo(VECT) = 0x 0000 0000 1000 0014 - 0x 0000 0000 0040 0014 = 0x 0000 0000 0FC0 0000 = 0x 000 (12 bit inf)	jal #pcrel(MAINEND)/2= (0x 0000 0000 0040 0024 - 0x 0000 0000 0040 0048)/2 = (0x FFFF FFFF FFFF FFDC)/2 = 0xFFFE E (20 bit)
beq #pcrel(MODULE)/2= (0x 0000 0000 0040 0034 - 0x 0000 0000 0040 0020)/2 = (0x 0000 0000 0000 0014)/2 = 0x00A (12 bit)	
jal #pcrel(MAIN)/2= (0x 0000 0000 0040 0000 - 0000 0000 0040 0024)/2 = (0x FFFF FFFF FFFF FFDC)/2 = 0x00FEE (20 bit)	

(5) – codice eseguibile

testo	
indirizzo	codice (con codici operativi e registri in forma simbolica)
0x 0000 0000 0040 0000	lui t0, 0x FFF0 B
0x 0000 0000 0040 0004	addi t0, 0xBCC
0x 0000 0000 0040 0008	auipc t2, 0x0FC0 0
0x 0000 0000 0040 000C	addi t2, 0xFF8
0x 0000 0000 0040 0010	sw t0, 0(t2)
0x 0000 0000 0040 0014	auipc t3, 0x0FC0 0
0x 0000 0000 0040 0018	addi t3, 0x000
0x 0000 0000 0040 001C	lw t1, 0(t3)
0x 0000 0000 0040 0020	beq t0, t0, 0x00A
0x 0000 0000 0040 0024	jal zero, 0xFFFFE E
0x 0000 0000 0040 0028	auipc t4, 0x0FC0 0
0x 0000 0000 0040 002C	addi t4, 0xFF8
0x 0000 0000 0040 0030	lw t4, 0(t4)
0x 0000 0000 0040 0034	auipc t5, 0x 0 FC00
0x 0000 0000 0040 0038	addi t5, 0x FF0
0x 0000 0000 0040 003C	lb t6, 0(t5)
0x 0000 0000 0040 0040	sub t6, t6, t4
0x 0000 0000 0040 0044	beq t6, zero, 0x FF2
0x 0000 0000 0040 0048	jal zero, 0x FFFE E