Assembly Program Tracing

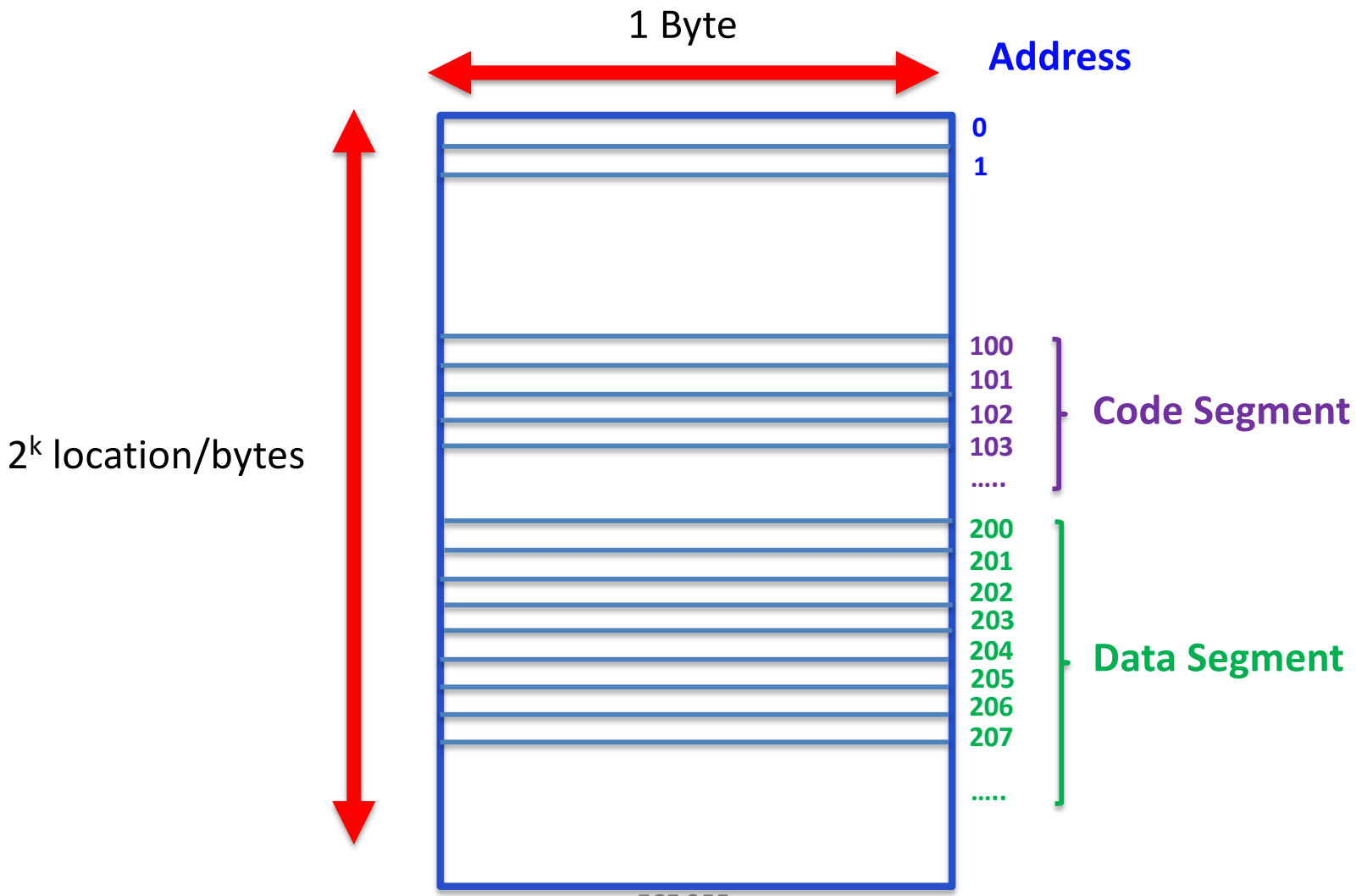
Topics

1. Code and Data Segments
2. Code and Data Addresses
3. Pseudo Instructions
4. Tracing Registers
5. Application of Sample Program

Assembly Program Example

	Memory address label	Operation	Addressing or data information
Assembler directive		ORIGIN	100
Statements that generate machine instructions		LD	R2, N
		CLR	R3
		MOV	R4, #NUM1
	LOOP:	LD	R5, (R4)
		ADD	R3, R3, R5
		ADD	R4, R4, #4
		SUB	R2, R2, #1
		BGT	R2, R0, LOOP
		ST	R3, SUM
		next instruction	
Assembler directives		ORIGIN	200
	SUM:	RESERVE	4
	N:	DATAWORD	150
	NUM1:	RESERVE	600
		END	

Code and Data Segments



Code and Data Addresses Exercise

	Memory address label	Operation	Addressing or data information
Assembler directive		ORIGIN	100
Statements that generate machine instructions		LD	R2, N
		CLR	R3
		MOV	R4, #NUM1
	LOOP:	LD	R5, (R4)
		ADD	R3, R3, R5
		ADD	R4, R4, #4
		SUB	R2, R2, #1
		BGT	R2, R0, LOOP
		ST	R3, SUM
		next instruction	
Assembler directives		ORIGIN	200
	SUM:	RESERVE	4
	N:	DATAWORD	150
	NUM1:	RESERVE	600
		END	

Code and Data Addresses

	Memory address label		Operation	Addressing or data information
Assembler directive			ORIGIN	100
Statements that generate machine instructions		100	LD	R2, N
		104	CLR	R3
		108	MOV	R4, #NUM1
	LOOP:	112	LD	R5, (R4)
		116	ADD	R3, R3, R5
		120	ADD	R4, R4, #4
		124	SUB	R2, R2, #1
		128	BGT	R2, R0, LOOP
		132	ST	R3, SUM
			next instruction	
Assembler directives			ORIGIN	200
	SUM:	200	RESERVE	4
	N:	204	DATAWORD	150
	NUM1:	208	RESERVE	600
		212	END	
		xxx		
		.....		
		804		

Pseudo Instructions Exercise

	Memory address label		Operation	Addressing or data information
Assembler directive			ORIGIN	100
Statements that generate machine instructions		100	LD	R2, N
		104	CLR	R3
		108	MOV	R4, #NUM1
	LOOP:	112	LD	R5, (R4)
		116	ADD	R3, R3, R5
		120	ADD	R4, R4, #4
		124	SUB	R2, R2, #1
		128	BGT	R2, R0, LOOP
		132	ST	R3, SUM
			next instruction	
Assembler directives			ORIGIN	200
	SUM:	200	RESERVE	4
	N:	204	DATAWORD	150
	NUM1:	208	RESERVE	600
		212	END	
		xxx		
		.....		
		804		

Pseudo Instructions

	Memory address label		Operation	Addressing or data information	
Assembler directive			ORIGIN	100	
Statements that generate machine instructions		100	LD	R2, N	R2 ← M[204]
		104	CLR	R3	R3 ← 0
		108	MOV	R4, #NUM1	R4 ← 208
	LOOP:	112	LD	R5, (R4)	R5 ← M[208]
		116	ADD	R3, R3, R5	R3 ← R3 + R5
		120	ADD	R4, R4, #4	R4 ← R4 + 4
		124	SUB	R2, R2, #1	R2 ← R2 - 1
		128	BGT	R2, R0, LOOP	If R2 > R0, PC=112
		132	ST	R3, SUM	R3 → M[200]
			next instruction		
Assembler directives			ORIGIN	200	
	SUM:	200	RESERVE	4	
	N:	204	DATAWORD	150	
	NUM1:	208	RESERVE	600	
		212	END		
		xxx			
		.....			
		804			

Given the following, find out what this program does:

Many Processors have R0 set to 0

Tracing: Sample Assembly Program

Memory address label		Operation	Addressing or data information	
		ORIGIN	100	
	100	LD	R2, N	R2 ← M[204]
	104	CLR	R3	R3 ← 0
	108	MOV	R4, #NUM1	R4 ← 208
LOOP:	112	LD	R5, (R4)	R5 ← M[208]
	116	ADD	R3, R3, R5	R3 ← R3 + R5
	120	ADD	R4, R4, #4	R4 ← R4 + 4
	124	SUB	R2, R2, #1	R2 ← R2 - 1
	128	BGT	R2, R0, LOOP	If R2 > R0, PC=112
	132	ST	R3, SUM	R3 → M[200]
		next instruction		
		ORIGIN	200	
SUM:	200	RESERVE	4	
N:	204	DATAWORD	150	
NUM1:	208	RESERVE	600	
	212	END		
	xxx			
	.....			
	804			

R2	R3	R4	R5

Assignment: Show the values of each register after the execution of each instruction

Address	R2	R3	R4	R5
100	150	X	X	X
104				
108				
112				
116				
120				
124				
128				
132				

After the execution of the first instruction, we have (X means don't know or don't care)

Compiler Output

- https://docs.oracle.com/cd/E19957-01/806-3567/cc_options.html
  - CC with option/flag -S to produce an assembly file
- https://www.delorie.com/djgpp/v2faq/faq8_20.html
  - GCC to generate assembly code
- https://godbolt.org/
  - Square of an integer: C to assembly (M68K, ARM-32, 13.2)
- https://www.codeconvert.ai/c-to-assembly-converter
  - C to assembly converter