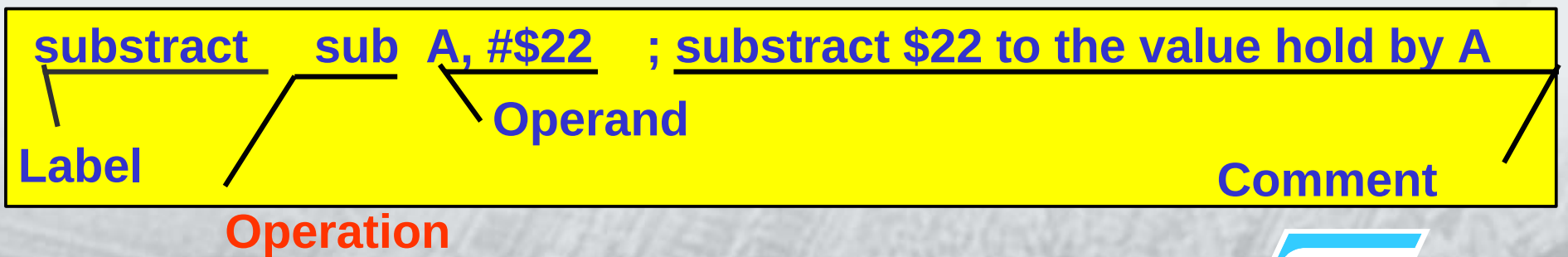
ST7 MICROCONTROLLER TRAINING

1. INTRODUCTION
2. CORE
3. ADDRESSING MODES
4. ASSEMBLY TOOLCHAIN
5. STVD7 DEBUGGER
6. HARDWARE TOOLS
7. PERIPHERALS
8. ST-REALIZER II
9. C TOOLCHAINS

SOFTWARE DESCRIPTION
Source Program Format

- A program is made up of lines. Each assembler line consists of up to 4 fields
  - **The label field: it is an identifying label**
  - **The operation field: it contains a word which specifies the action to be performed by the assembly process**
  - **The operand field: it is constitued of quantity (number or register) on which the operation is performed**
  - **The comment field: for any comment concerning the line**



INHERENT

- The OP code fully specify all required information for the cpu to process the operation (The inherent instructions are single byte long).

- Example

  - **RCF - Reset Carry Flag ; RSP - Reset Stack Pointer**
  - **HALT ; WFI - Low consumption modes**
  - **TRAP - Enter software interrupt**
  - **RET - Return ; IRET -Interrupt Return**
  - **SIM - Set Interrupt Mask ; RIM - Reset Interrupt Mask**
  - **PUSH A ; POP Y - Stack operation**

IMMEDIATE ADDRESSING MODE

- **BCP A, #$FF**
- **XOR A, #%01001110**
- **LD X, #255**

- Example

Before Completion

A = Previous value

LD A, #$55

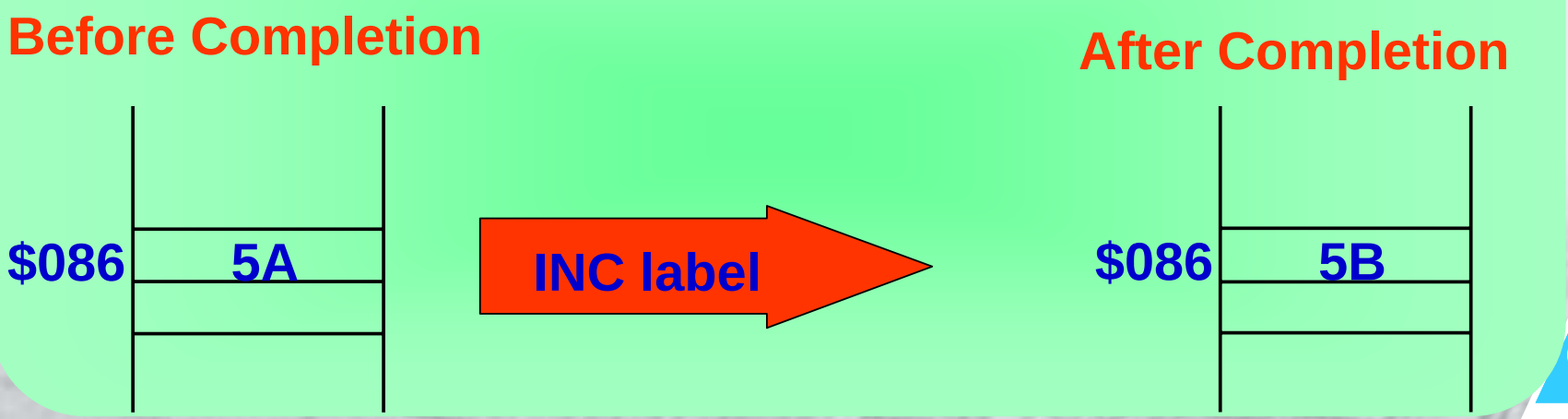
After Completion

A = $55

SHORT DIRECT ADDRESSING MODE

- ADDRESSABLE SPACE: 00 to FF
  - **INC variable**
  - **NEG variable**
  - **SRA variable**

- Example
  - **If label is an 8 bit symbol defined as $86**



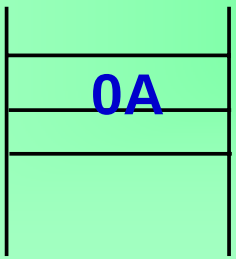
LONG DIRECT ADDRESSING MODE

- ADDRESSABLE SPACE: 0000 to FFFF
  - **CP Y, VARIABLE**
  - **SBC A, VARIABLE**

- Example
  - **If label is an 16 bit symbol defined as $F1D6**

Before Completion

A = $C1



ADD A, LABEL

After Completion

A = $CB

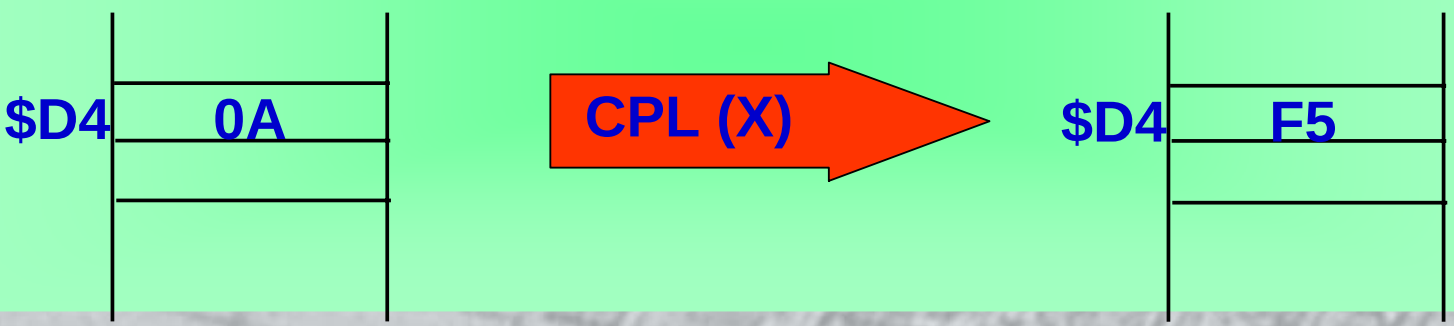
NO OFFSET INDEXED ADDRESSING MODE

- ADDRESSABLE SPACE: 00 to FF
  - **CPL (X)**
  - **DEC (Y)**
  - **OR A, (X)**

- Example

Before Completion

X contains $D4



After Completion

SHORT INDEXED ADDRESSING MODE

- ADDRESSABLE SPACE : 000 to 1FE
  - **RRC (variable, X)**
  - **SWAP (variable, Y)**

- Example
  - **If label is an 8 bit symbol defined as $90**

Before Completion
X = 03 and C=1

$90	0A
$91	1E
$92	F6
$93	A8

RRC (label, X)

After Completion
C=0

$90	0A
$91	1E
$92	F6
$93	D4

LONG INDEXED ADDRESSING MODE

- ADDRESSABLE SPACE: 0000 to FFFF
  - **ADD A, (variable, Y)**
  - **XOR A, (variable, X)**

- Example
  - **If label is an 16 bit symbol defined as $2E6D**

Before Completion

X = 02

$2E6D	**0A**
$2E6E	**1E**
$2E6F	**2F**
$2E70	**A8**

LD X, (LABEL, X)

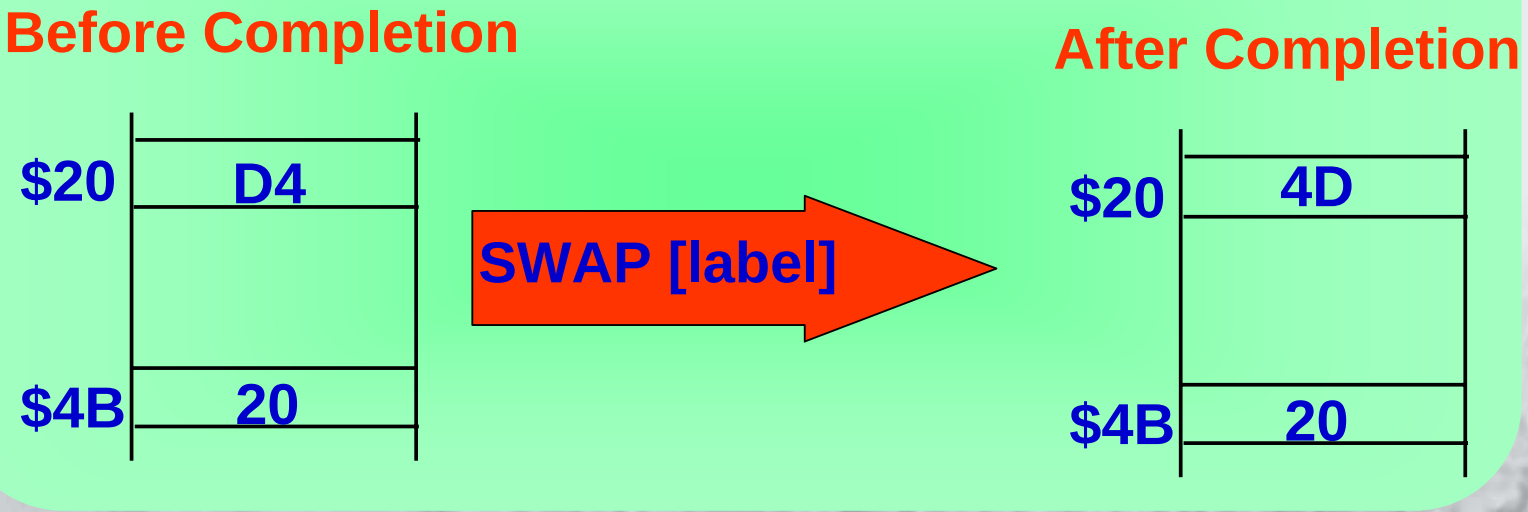
After Completion

X = $2F

SHORT INDIRECT ADDRESSING MODE

- ADDRESSABLE SPACE: 00 to FF
  - **TNZ [variable]**
  - **CLR [variable]**

- Example
  - **If label is an 8 bit symbol defined as $4B**

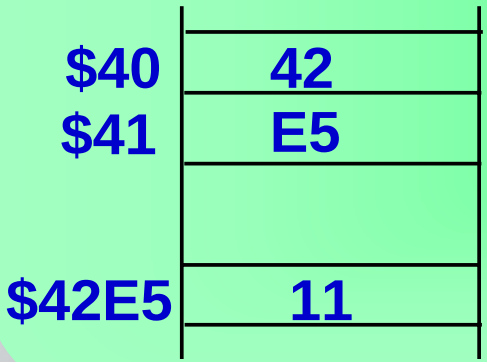


LONG INDIRECT ADDRESSING MODE

- ADDRESSABLE SPACE: 0000 to FFFF
  - **ADC A, [variable.w]**
  - **BCP A, [variable.w]**

- Example
  - **If label is an 8 bit symbol defined as $40**

Before Completion



LD A, [label.w]

After Completion

A = $11

SHORT INDIRECT INDEXED

- ADDRESSABLE SPACE: 000 to 1FE
  - **RLC ([variable], Y)**
  - **LD X, ([variable], X)**

- Example
  - **If label is an 8 bit symbol defined as $C8**

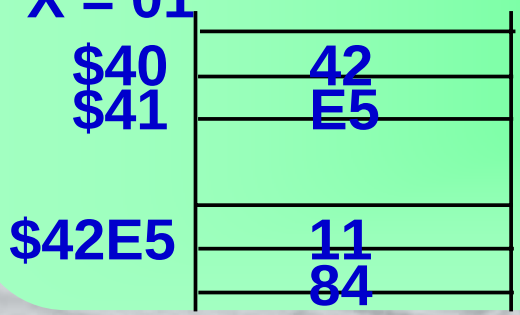
LONG INDIRECT INDEXED

- ADDRESSABLE SPACE : 0000 to FFFF
  - **SUB A, ([variable.w], X)**
  - **AND A, ([variable.w], Y)**

- Example
  - **If label is an 8 bit symbol defined as $40**

Before Completion

X = 01



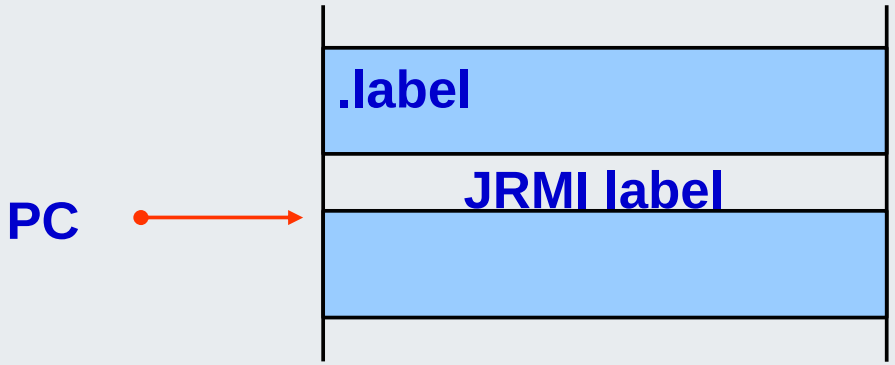
LD A, ([label.w],X)

After Completion

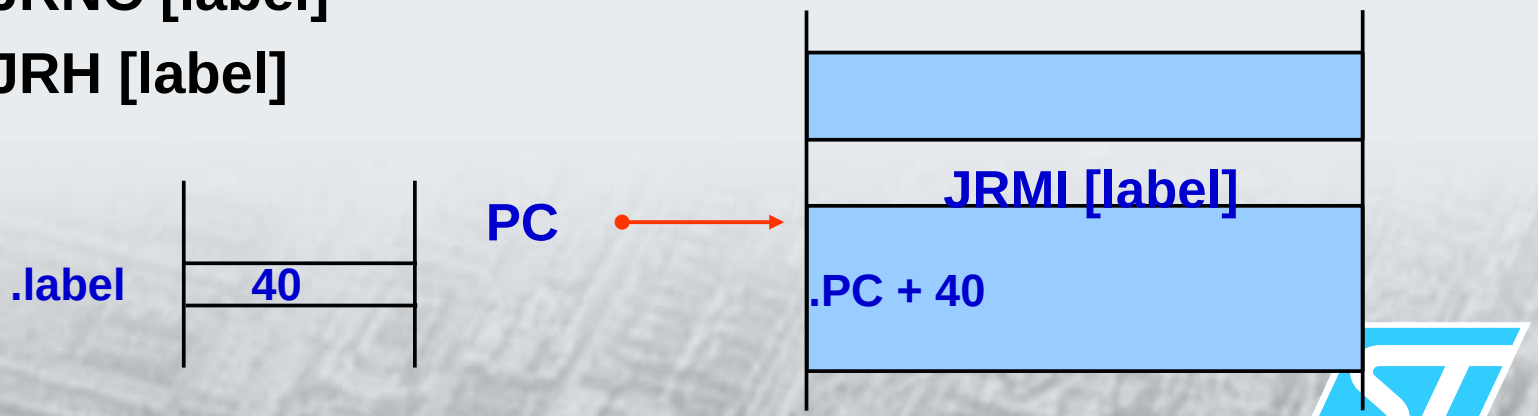
A = $84

RELATIVE ADDRESSING MODE

- DIRECT ADDRESSABLE SPACE: PC-128 to PC+127
  - **JRUGE label**
  - **CALLR label**



- INDIRECT ADDRESSABLE SPACE: PC-128 to PC+127
  - **JRNC [label]**
  - **JRH [label]**



BIT MANIPULATION

- DIRECT ADDRESSABLE SPACE: 00 to FF

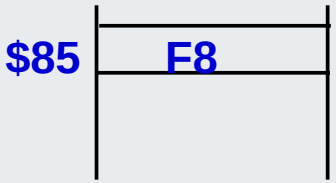
- INDIRECT ADDRESSABLE SPACE: 00 to FF

  variable = $86

RELATIVE JUMP ON BIT TEST

- DIRECT ADDRESSABLE SPACE: 00 to FF

variable address : $85

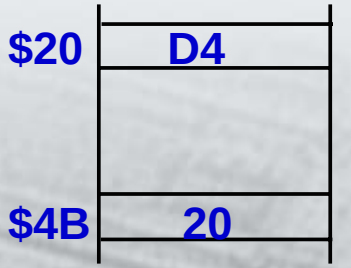


BTJT variable, #3, label

Relative jump to label in range PC-128 to PC+127

- INDIRECT ADDRESSABLE SPACE: 00 to FF

variable address : $4B

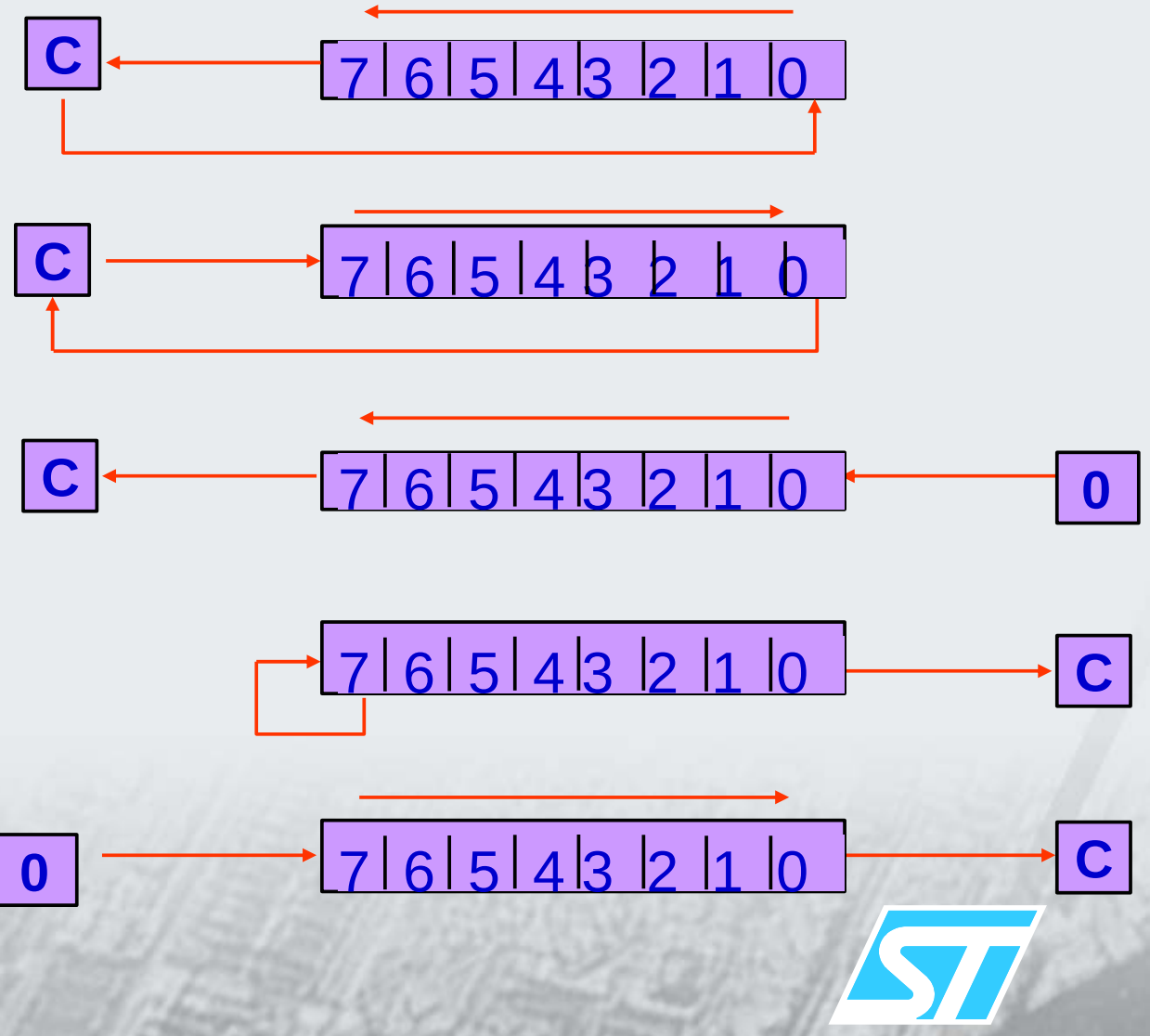


BTJF [variable], #7, label

Execute next instruction

SHIFT AND ROTATE INSTRUCTIONS

- RLC
- RRC
- SLA and SLL
- SRA
- SRL



ADDRESSING MODES Summary

ADDRESSING MODE	SYNTAX EXAMPLES	ADDRESS RANGE
Inherent	RCF ; MUL X,A ; SWAP A ; CLR X ; WFI	Not applicable
Immediate	BCP A, #$0FF ; XOR A, #%01001110	Not applicable
Short Direct	INC variable ; NEG variable ; SRA variable	00...FF
Long Direct	CP Y, VARIABLE ; SBC A, VARIABLE	0000...FFFF
No offset Indexed	CPL (X) ; DEC (Y) ; OR A, (X)	00...FF
Short Indexed	RRC (variable, X) ; SWAP (variable, Y)	000...1FE
Long Indexed	ADD A, (VARIABLE, Y)	0000...FFFF
Short Indirect	TNZ [variable]	00...FF
Long Indirect	ADC A, [variable.w]	0000...FFFF
Short Indirect Indexed	RLC ([variable], Y) ; LD X, ([variable], X)	000...1FE
Long Indirect Indexed	SUB A, ([variable.w], X)	0000...FFFF
Relative Direct	JRUGE label ; CALLR label	PC -128/+127
Relative Indirect	JRNC [label] ; JRH [label]	PC -128/+128
Bit Direct	BRES variable, #7	00...FF
Bit Indirect	BSET [variable], #7	00...FF
Relative Bit Direct	BTJT variable, #3, label	00...FF
Relative Bit Indirect	BTJF [variable], #7, label	00...FF

EXERCICES ON ADDRESSING MODES

- What is the content of the registers X,Y and A after the execution of the 3 instructions ?

Before Completion

X = 03 and Y = 04

```
LD   X,(Label,X)
LD   Y,([Label],Y)
LD   A,([Label2.w],Y)
```

After Completion

X = ?
Y = ?
A = ?

	Memory Address	Memory Content
Label2:	$009F	OO
	$00A0	A6
	$00A1	53
Label:	$00A2	A0
	$00A3	B8
	$00A4	O1
	$00A5	O4
	$00A6	D3
	$00A7	61