



**ECE375**

# **Large Number Arithmetic**

**TA:**

School of Electrical Engineering and Computer Science  
Oregon State University



# Goal of this Lab

- Build arithmetic functions.
- Manipulate large numbers in Assembly.
- Properly use subroutines.
- Use memory windows to observe data.

# Large Number Arithmetic

- Assembly use 8bit instruction set.
- Figure out how to build arithmetic operations on numbers that are larger than 8 bits
- Operands are provided in skeleton code.

# 16bit Adder

+		\$A2	\$FF
		\$F4	\$77
<hr/>			
	\$01	\$97	\$76

+	\$0101 : \$0100
	\$0103 : \$0102
<hr/>	
	\$0106 : \$0105 : \$0104

# Program Memory to Data Memory

## Program Memory

\$A2	\$FF	\$0200
\$F4	\$77	\$0201
\$00	\$00	\$0202

## Data Memory

\$D1	\$0100
\$D2	\$0101
\$D3	\$0102
\$D4	\$0103

# Program Memory to Data Memory



**Program Memory**

\$A2	\$FF	\$0200
\$F4	\$77	\$0201
\$00	\$00	\$0202

**Data Memory**

<b>\$FF</b>	\$0100
<b>\$A2</b>	\$0101
<b>\$77</b>	\$0102
<b>\$F4</b>	\$0103

# Program Memory to Data Memory

Program Memory

\$A2	\$FF	\$0200
\$F4	\$77	\$0201
\$00	\$00	\$0202

Data Memory

<b>\$FF</b>	\$0100
<b>\$A2</b>	\$0101
<b>\$77</b>	\$0102
<b>\$F4</b>	\$0103

	\$A2	\$FF
	\$F4	\$77
	<hr/>	
\$01	\$97	\$76

Operations

**ADD**

**SUB**

**MUL**



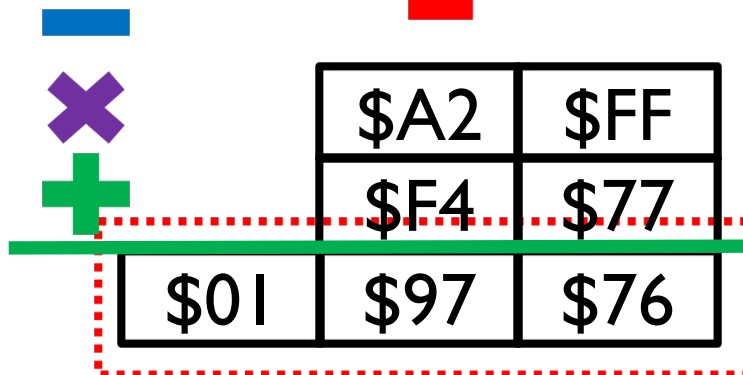
# Program Memory to Data Memory

Program Memory

\$A2	\$FF	\$0200
\$F4	\$77	\$0201
\$00	\$00	\$0202

Data Memory

\$FF	\$0100
\$A2	\$0101
\$77	\$0102
\$F4	\$0103
\$76	\$0104
\$97	\$0105
\$01	\$0106





# 16bit Adder

FUNCTION:

ldi XL, \$00

ldi XH, \$01

ldi YL, \$02

ldi YH, \$01

ldi ZL, \$04

ldi ZH, \$01

\$A2	\$FF
\$F4	\$77
\$00	\$00
\$00	\$00

\$0101 : \$0100

\$0103 : \$0102

\$0105 : \$0104

\$0107 : \$0106



# 16bit Adder

FUNCTION:

ldi XL, \$00

ldi XH, \$01

ldi YL, \$02

ldi YH, \$01

ldi ZL, \$04

ldi ZH, \$01

\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$00	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

X

Y

Z

# 16bit Adder

FUNCTION:

ldi XL, \$00

ldi XH, \$01

ldi YL, \$02

ldi YH, \$01

ldi ZL, \$04

ldi ZH, \$01

\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$00	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106



# 16bit Adder

FUNCTION:

ldi XL, \$00

ldi XH, \$01

ldi YL, \$02

ldi YH, \$01

ldi ZL, \$04

ldi ZH, \$01

\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$00	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106



# 16bit Adder

```
ld    R16,X+
ld    R17,Y+
add   R17,R16
st    Z+,R17
ld    R16,X
ld    R17,Y
adc   R17,R16
st    Z+,R17
brcc  EXIT
st    Z,XH
EXIT:
      ret
```

\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$00	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$16	R16
\$17	R17
C=0	SREG

# 16bit Adder

**ld R16, X+**

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

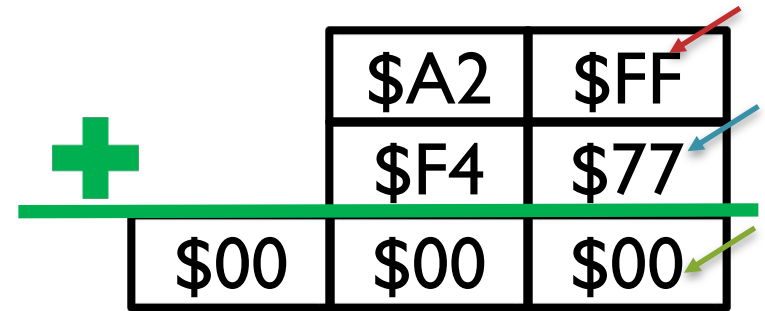
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	<b>\$FF</b>	\$0101 : \$0100
\$F4	<b>\$77</b>	\$0103 : \$0102
\$00	<b>\$00</b>	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

<b>\$FF</b>	R16
<b>\$17</b>	R17
<b>C=0</b>	SREG

# 16bit Adder

**ld R16, X+**

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

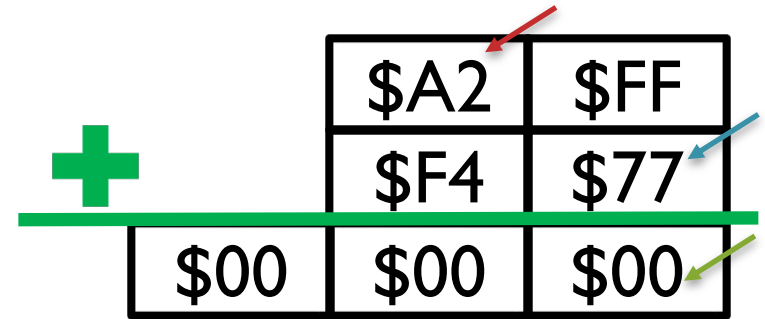
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$00	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$FF	R16
\$17	R17
C=0	SREG



# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

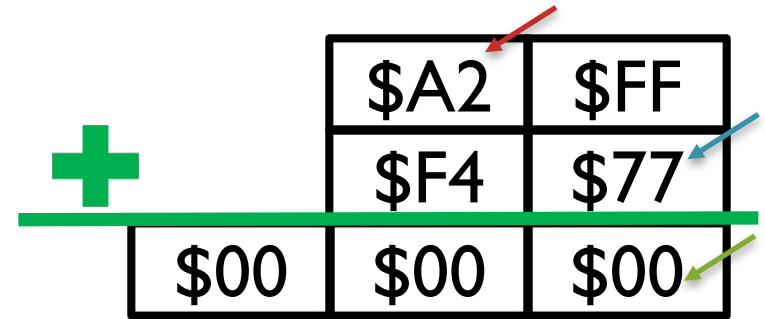
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$00	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$FF	R16
\$77	R17
C=0	SREG

# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

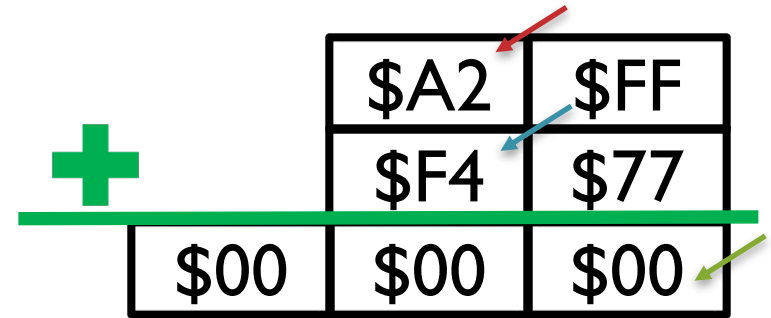
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$00	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$FF	R16
\$77	R17
C=0	SREG

# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

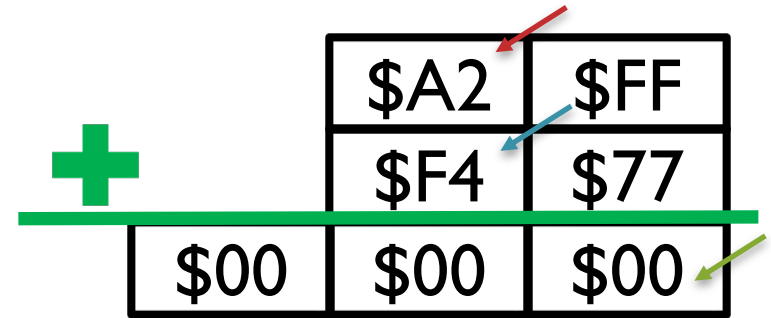
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$00	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$FF	R16
\$77	R17
C=0	SREG

# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

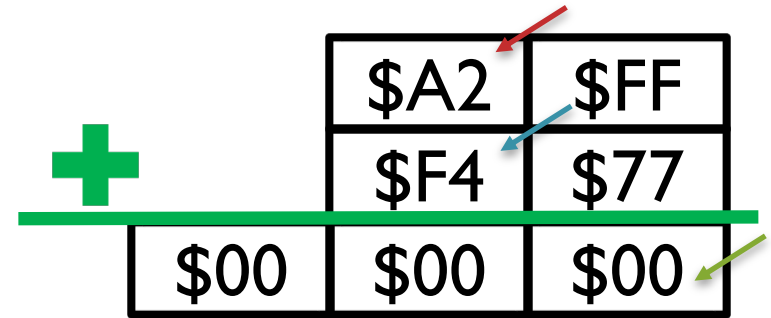
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$00	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$FF	R16
\$76	R17
C=1	SREG

# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

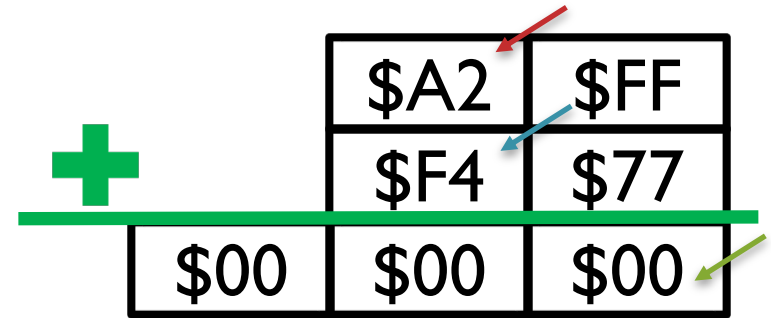
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$00	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$FF	R16
\$76	R17
C=1	SREG

# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

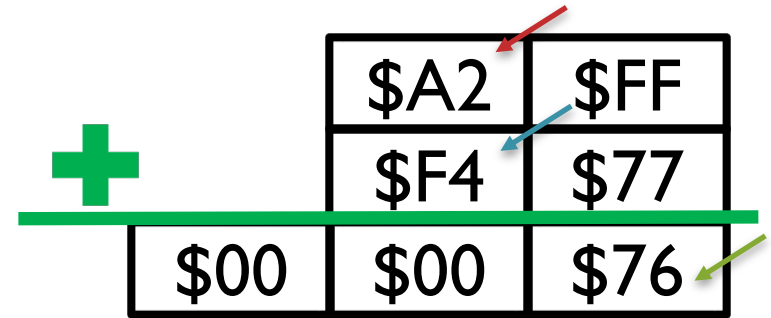
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$76	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$FF	R16
\$76	R17
C=1	SREG

# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

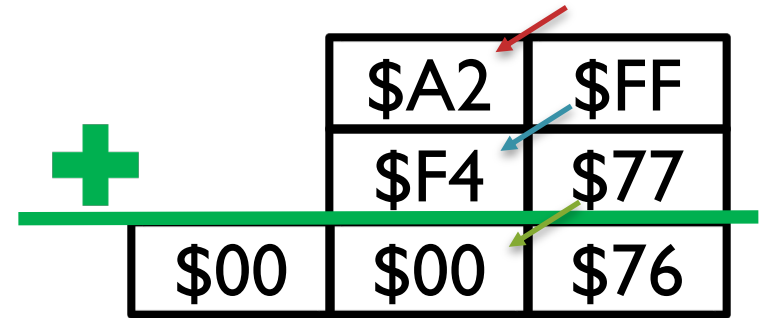
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$76	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$FF	R16
\$76	R17
C=1	SREG



# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

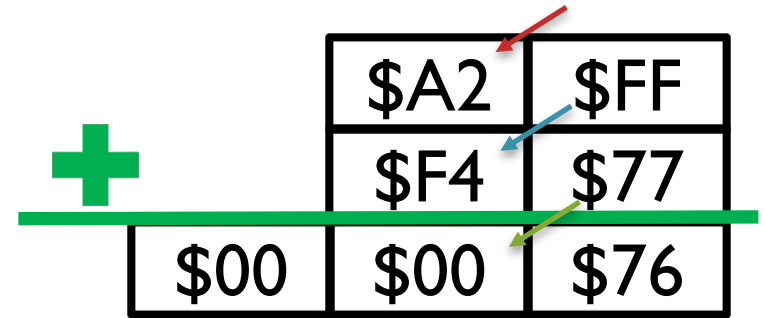
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$76	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$A2	R16
\$76	R17
C=1	SREG

# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

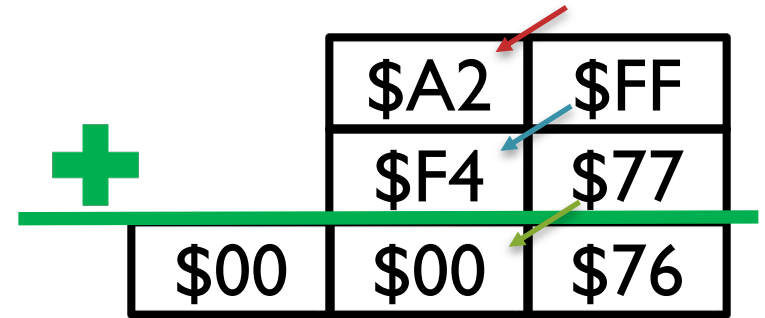
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$76	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$A2	R16
\$F4	R17
C=1	SREG

# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

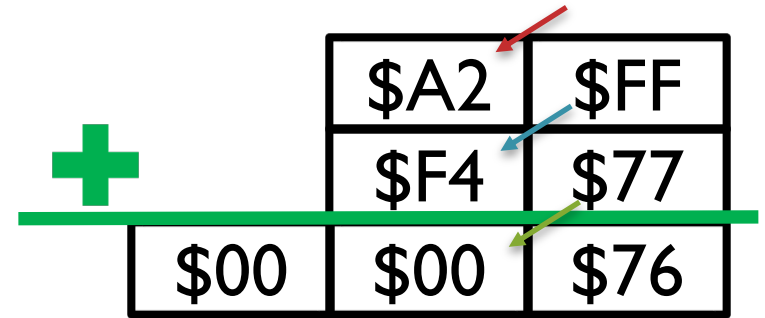
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$76	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$A2	R16
\$97	R17
C=1	SREG

# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

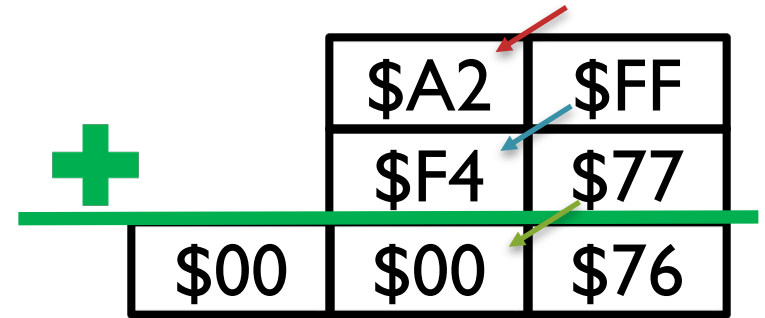
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$00	\$76	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$A2	R16
\$97	R17
C=1	SREG

# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

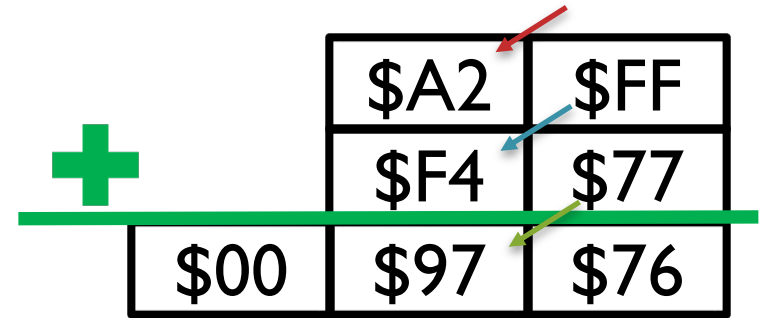
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$97	\$76	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$A2	R16
\$97	R17
C=1	SREG

# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

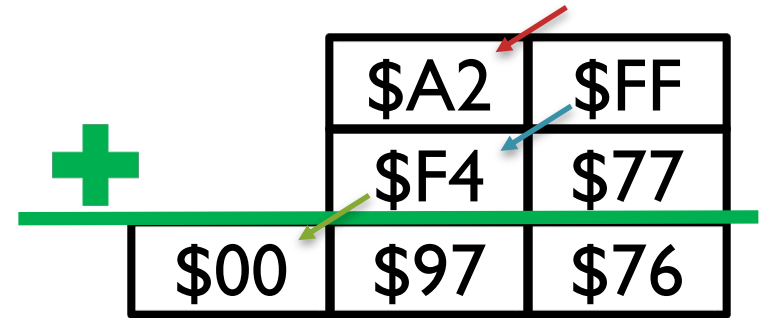
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$97	\$76	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$A2	R16
\$97	R17
C=1	SREG

# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

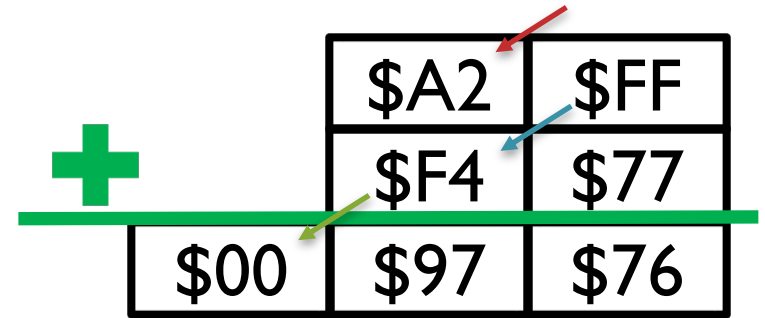
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$97	\$76	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$FF	R16
\$76	R17
C=1	SREG



# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

st Z+, R17

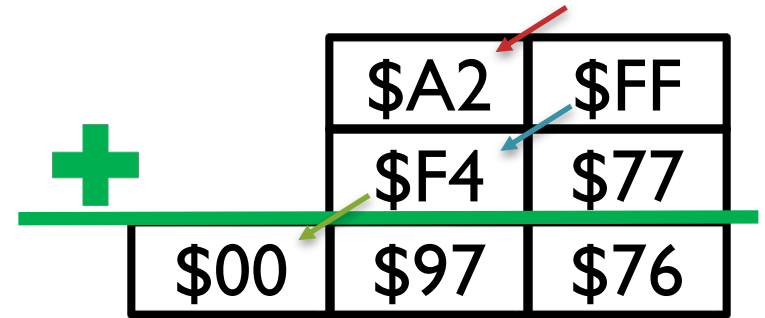
brcc EXIT

st Z, XH

EXIT:

ret

ldi R18, \$01  
st. Z, R18



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$97	\$76	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106

\$FF	R16
\$76	R17
C=1	SREG

# 16bit Adder

ld R16, X+

ld R17, Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17, Y

adc R17, R16

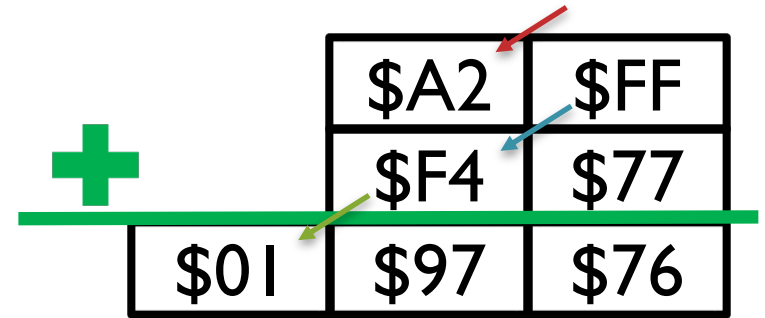
st Z+, R17

brcc EXIT

st Z, XH

EXIT:

ret



\$A2	\$FF	\$0101 : \$0100
\$F4	\$77	\$0103 : \$0102
\$97	\$76	\$0105 : \$0104
\$00	\$01	\$0107 : \$0106

\$FF	R16
\$76	R17
C=1	SREG

ldi R18, \$01  
st. Z, R18

# Check-off Lists

- Correctly defined operands.
- Correct data manipulation for operands.
- Correct results for each arithmetic function.

# Announcements

- We don't use AVR board for this lab.
- You must use Windows to complete this lab assignments.

# Questions?

