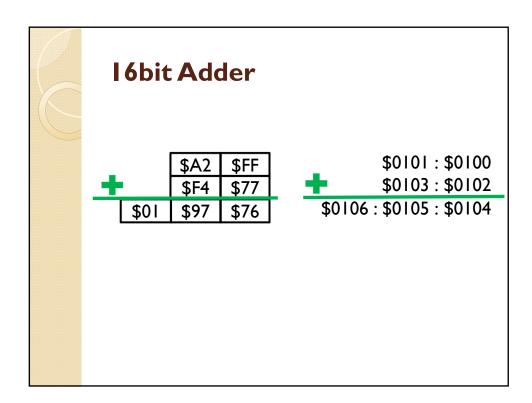
ECE375 Lab 5

TA: Youngbin Jin

School of Electrical Engineering and Computer Science Oregon State University

Large Number Arithmetic

- AVR 8-bit instruction set
- Perform arithmetic operations on numbers that are larger than 8 bits
- Practice Memory Manipulation
- The Skeleton Code includes operands for ADD16, SUB16, MUL24, and Compound



FUNCTION:

ldi XL, \$00

ldi XH, \$01

ldi YL, \$02

ldi YH, \$01

ldi ZL, \$04

Idi ZH, \$01

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







FUNCTION:

XL, \$00 ldi

XH, \$01 ldi

YL, \$02 ldi

YH, \$01 ldi

ZL, \$04 ldi

ZH, \$01 ldi

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







I 6bit Adder

FUNCTION:

XL, \$00 ldi

XH, \$01 ldi

YL, \$02 ldi

YH, \$01 ldi

ZL, \$04 ldi

ZH, \$01 ldi

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







FUNCTION:

XL, \$00 ldi

XH, \$01 ldi

YL, \$02 ldi

YH, \$01 ldi

ZL, \$04 ldi

ZH, \$01 ldi

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







I 6bit Adder

FUNCTION:

XL, \$00 ldi

XH, \$01 ldi

YL, \$02 ldi

YH, \$01 ldi

ZL, \$04 ldi

ldi

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

ZH, \$01







ld R16, X+ R17,Y+ ld R17, R16 add Z+, R17 st R16, X ld **R17,Y** ld R17, R16 adc Z+, R17 st

brcc EXIT st Z, XH EXIT:

\$16 \$17 RI6 RI7 SREG

ret

I 6bit Adder

Ιd R16, X+ Ιd R17,Y+ R17, R16 add Z+, R17 st R16, X ld **R17,Y** ld R17, R16 adc Z+, R17 st brcc EXIT Z, XH st

ret

EXIT:

\$FF \$17 C=0

RI6 RI7 SREG

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



I 6bit 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 \$0103 : \$0102
\$F4	\$77	\$0103 : \$0102
\$00	\$00	\$0105 : \$0104
\$00	\$00	\$0107 : \$0106



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

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

R16 \$FF RI7 **SREG**

I 6bit Adder

R16, X+ R17,Y+ Ιd RI7, RI6 add Z+, R17 st R16, X ld **R17,Y** ld R17, R16 adc Z+, R17 st

ld

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

brcc EXIT Z, XH st EXIT:

ret

\$FF **R16 R17 SREG**

Id 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

St ∠,∧⊓

EXIT:

ret

\$A2 \$FF \$0 \$F4 \$77 \$0

\$0101 : \$0100

\$F4 \$77 \$0103 : \$0102 \$00 \$00 \$0105 : \$0104

\$00 \$0107 : \$0106

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

\$00

I 6bit Adder

ld R16, X+

ld R17,Y+

add R17, R16

st Z+, R17

ld R16, X

ld R17,Y

adc R17, R16

adc 1(17,1(10

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

ld R16, X+ R17,Y+ ld

R17, R16 add

Z+, RI7 st

R16, X ld

R17,Y ld

R17, R16 adc

Z+, R17 st

brcc EXIT

Z, XH st

EXIT:

ret

\$FF \$0101:\$0100 \$A2 \$F4 \$77

\$0103:\$0102 \$0105 : \$0104 \$76 \$00

\$0107 : \$0106 \$00 \$00

\$FF **R16** RI7 \$76

SREG

I 6bit Adder

R16, X+ ld

R17,Y+ Ιd

R17, R16 add

Z+, R17 st

R16, X ld

R17,Y Ιd

R17, R16 adc

Z+, R17 st

brcc EXIT

Z, XH st

EXIT:

ret

\$0101:\$0100 \$A2 \$FF \$0103:\$0102 \$F4 \$77

\$00 \$76 \$0105 : \$0104

\$0107 : \$0106 \$00 \$00

\$FF **R16 R17** \$76 **SREG**

ld R16, X+ R17,Y+ ld R17, R16 add Z+, R17 st

RI6,X Ιd

R17,Y ld R17, R16 adc Z+, R17 st brcc EXIT

Z, XH st

EXIT:

ret

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

RI6 R17 \$76 **SREG**

I 6bit Adder

R16, X+ ld R17,Y+

Ιd

R17, R16 add Z+, R17 st

R16, X ld

RI7,Y ld

R17, R16 adc

Z+, R17 st

brcc EXIT

Z, XH st

EXIT:

ret

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

\$0107 : \$0106 \$00 \$00

R16 R17 SREG

ld R16, X+ ld R17,Y+ R17, R16 add Z+, R17 st R16, X ld **R17,Y** ld R17, R16 adc

Z+, R17 st

brcc EXIT

Z, XH st EXIT:

ret

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

\$A2

RI7 **SREG**

RI6

I 6bit Adder

R16, X+ ld Ιd R17,Y+ R17, R16 add Z+, R17 st ld R16, X

R17,Y ld R17, R16 adc

Z+, R17 st

EXIT brcc

Z, XH st

EXIT:

ret

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

\$0107 : \$0106 \$00 \$00

R16 \$A2 **R17** \$97 **SREG**

ld R16, X+ R17,Y+ ld R17, R16 add Z+, R17 st R16, X ld **R17,Y** ld R17, R16 adc Z+, R17 st

 \$A2
 \$FF
 \$0101 : \$0100

 \$F4
 \$77
 \$0103 : \$0102

 \$97
 \$76
 \$0105 : \$0104

 \$00
 \$0107 : \$0106

brcc EXIT
st Z, XH
EXIT:
ret

\$A2 \$97

RI6 RI7 SREG

I 6bit Adder

Id R16, X+
Id R17, Y+
add R17, R16
st Z+, R17
Id R16, X
Id R17, Y
adc R17, R16

 \$A2
 \$FF
 \$0101 : \$0100

 \$F4
 \$77
 \$0103 : \$0102

 \$97
 \$76
 \$0105 : \$0104

 \$00
 \$0107 : \$0106

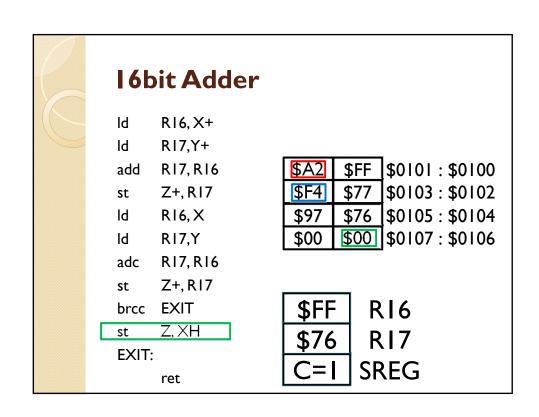
st Z+, R17

brcc EXIT st Z, XH EXIT:

\$A2 \$97 RI6 RI7 SREG

ret

16bit Adder ld R16, X+ R17,Y+ ld \$FF \$0101:\$0100 R17, R16 \$A2 add \$0103:\$0102 Z+, R17 \$F4 \$77 st \$0105 : \$0104 R16, X \$76 \$97 ld \$00 \$0107 : \$0106 **R17,Y** \$00 ld R17, R16 adc Z+, R17 st \$FF **R16** brcc EXIT Z, XH st **R17** \$76 EXIT: **SREG** ret



ld R16, X+ R17,Y+ ld R17, R16 add Z+, R17 st R16, X ld **R17,Y** ld R17, R16 adc Z+, R17 st brcc EXIT st Z, XH

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

EXIT:

ret

\$76

\$FF

R16 RI7 **SREG**

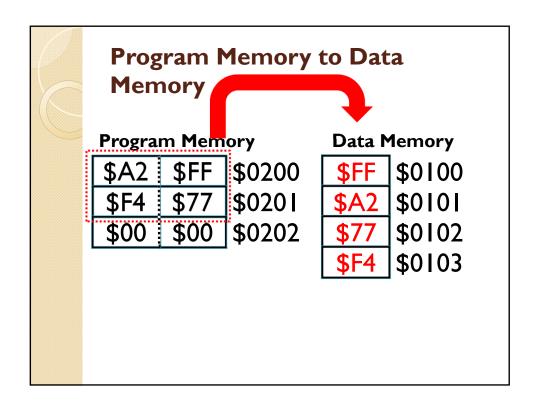
Program Memory to Data Memory

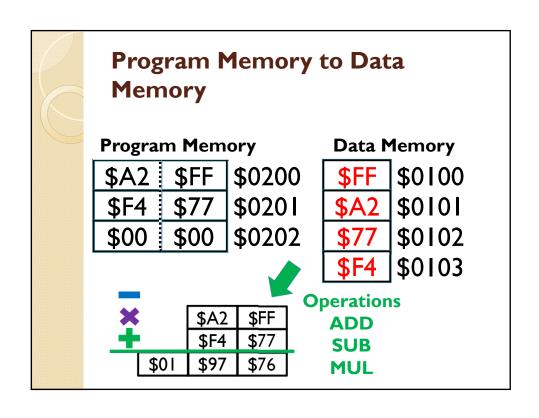
Program Memory

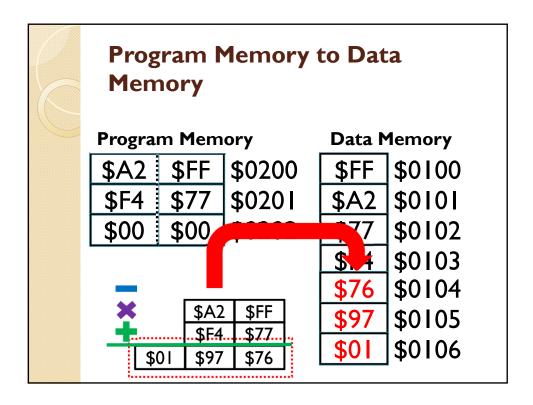
\$0200 \$A2 | \$FF \$020I \$77 \$F4 \$00 \$00 \$0202

Data Memory

\$DI \$0100 \$0101 \$D2 \$0102 \$D3 \$0103







Demo Check

- Insert Break Points
 - ADDI6
 - SUB16
 - MUL24
 - COMP
 - DONE
- TA will only check these break points
- You cannot modify memory during Demo

Checklist for Lab 5

- Demo Checklist
 - Correct ADD16 result w/ direct operands
 - Correct SUB16 result w/ direct operands
 - Correct MUL24 result w/ direct operands
 - Correct $((D E) + F)^2$ result
 - Good explanation of changes required to make MUL16 into a working MUL24?
- Challenge Checklist
 - MUL24 implemented using shift-and-add, and used for $((D E) + F)^2$ calculation

Questions?

