ECE375 Data Manipulation and the LCD

TA:

School of Electrical Engineering and Computer Science Oregon State University

Goal of this Lab

Learn data manipulation in Assembly.

Use X/Y/Z registers as pointers.

Use LCD driver function.

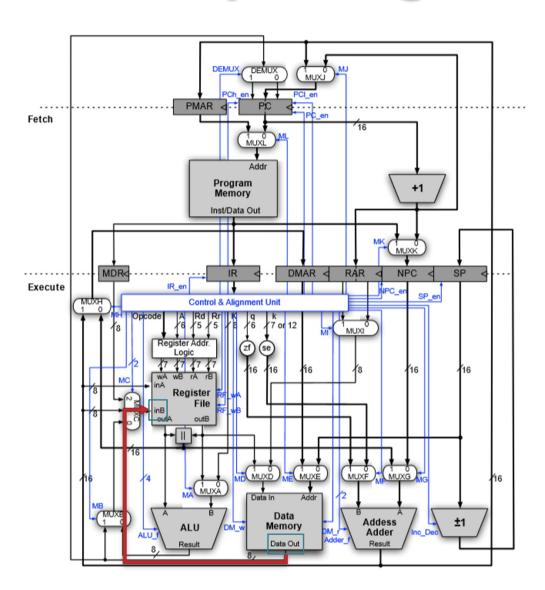
Data Manipulation

Data memory – Register

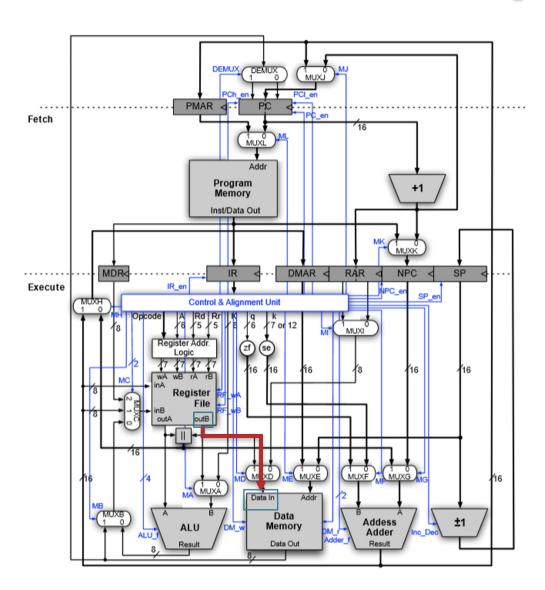
Data memory – Program Memory

Register – Program Memory

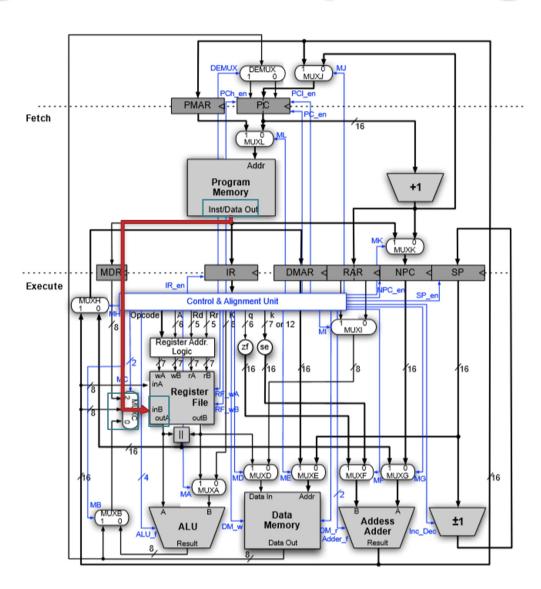
Data memory -> Register



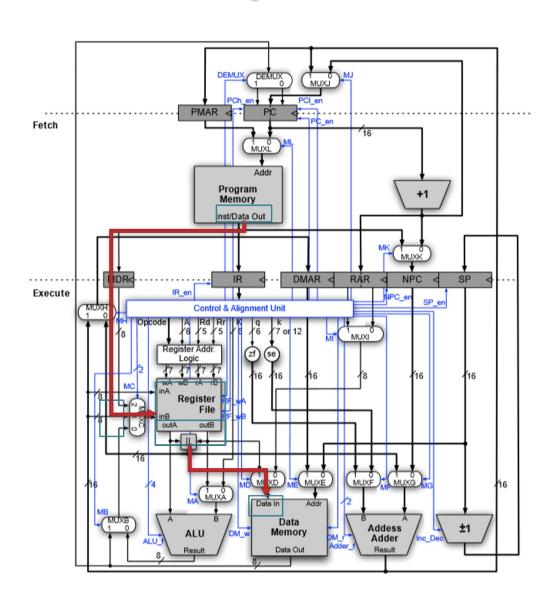
Register -> Data memory



Program memory -> Register



Program memory -> Data memory



Data Memory & Register

\$0100 \$0101 \$0102 \$0103 \$0104 \$0105 \$0106 8 Bit

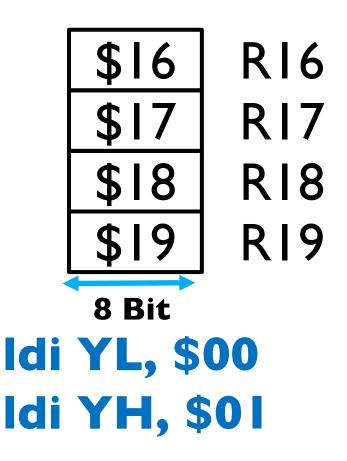
\$16 \$17 \$17 \$18 \$18 \$19 R19

Data Memory to Register

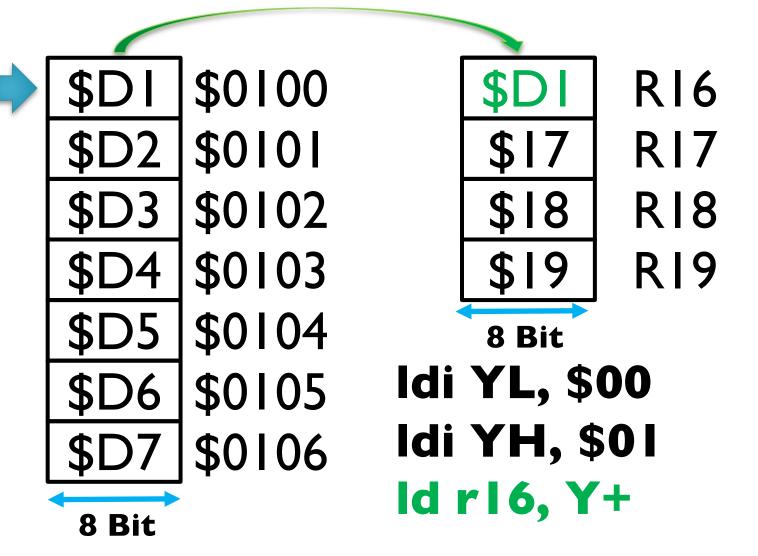


\$DI	\$0100
\$D2	\$0101
\$D3	\$0102
\$D4	\$0103
\$D5	\$0104
\$D6	\$0105
\$D7	\$0106

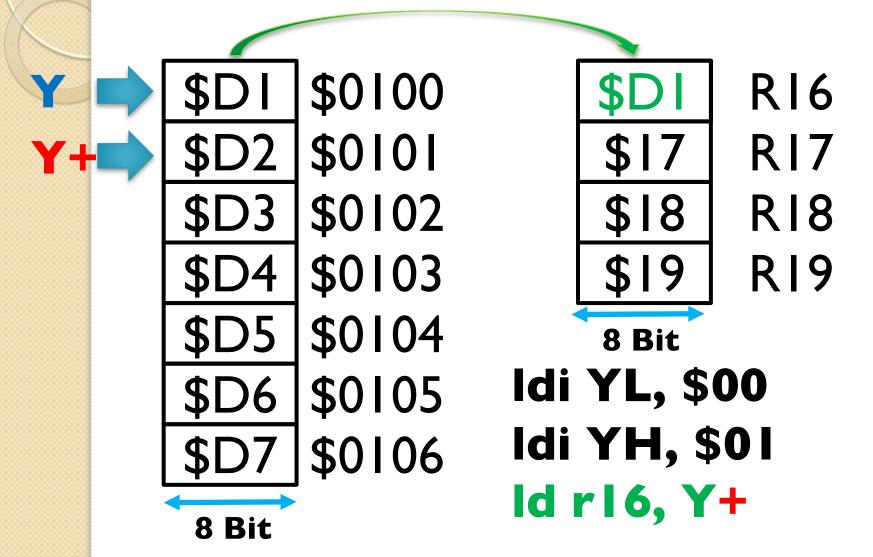
8 Bit



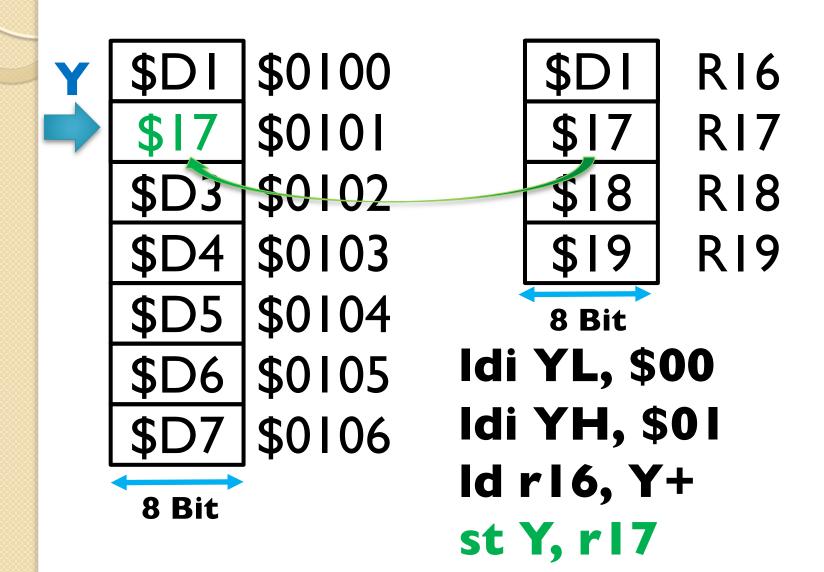
Data Memory to Register



Data Memory to Register



Register to Data Memory



•	16	Bit	•	8 Bit	•
	\$A2	\$AI	\$01FF	\$00	RI4
	\$A4	\$A3	\$0200	\$01	RI5
	\$A6	\$A5	\$020 I	\$16	RI6
	\$A8	\$A7	\$0202	\$17	RI7
	\$AA	\$A9	\$0203	ldi ZL	., \$00
	\$AC	\$AB	\$0204	ldi Zh	1, \$02
	\$AE	\$AD	\$0205		

743

Program Memory(16 bits) ≠ Register (8bits)

Cannot move data directly

16 Bit

\$A2:\$A1

\$A4: \$A3

\$A6: \$A5

\$A8: \$A7

\$AA: \$A9

\$AC: \$AB

\$AE:\$AD

\$01FF

\$0200

\$020I

\$0202

\$0203

\$0204

\$0205

8 Bit

\$00

\$01

\$16

\$17

R14

R15

R16

RI7

Idi ZL, \$00

Idi ZH, \$02

0 ZH 0 ZL 0

ZH (r31) ZL (r30)
0x0200 0

Z= 0b0000001000000000

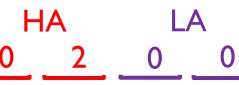
	D:4
10	BIT

\$A2	\$AI
\$ Z	\$A3
\$A6	\$A5
\$A8	\$A7
\$AA	\$A9
\$AC	\$AB
\$AE	\$AD

```
$01FF
$0200
$020 I
$0202
$0203
$0204
$0205
```

8 Bit	•
\$00	RI4
\$01	RI5
\$16	RI6
\$17	RI7
ldi ZL	. \$00





Z= 0b000001000000000high/low

6	Bit

$C \Lambda D$	•	ΦΛΙ
\$A2	-	DAI

\$	()	F	F
Y			•

Φ	\cap	7	\cap	\cap
\$	U		U	U

8 Bit

\$00	

Idi ZL, \$01

Idi ZH, \$04

Z= 0b000001000000001 high/low

16 Bit

\$A2 <u>\$A1</u> \$**Z** \$A3

\$A6: \$A5

\$A8: \$A7

\$AA: \$A9

\$AC: \$AB

\$AE:\$AD

\$01FF

\$0200

\$020 I

\$0202

\$0203

\$0204

\$0205

Idi **ZL**, \$00 Idi **ZH**, \$02

ldi ZL, (\$00<<1)

ldi ZH, (\$02<<1)

or

Idi ZL, \$00 Idi ZH, \$04

16 Bit

\$A2:\$AI

\$A6: \$A5

\$A8: \$A7

\$AA: \$A9

\$AC: \$AB

\$AE:\$AD

\$01FF

\$0200

\$020I

\$0202

\$0203

\$0204

\$0205

8 Bit

\$00

\$0I

\$16

\$17

R14

R15

R16

RI7

ldi ZL, \$00<<1

Idi ZH, \$02<<1

lpm r16, z+

lpm r17, z+

16 Bit

\$01FF \$A2: \$A \$0200 \$A6 : \$A5 \$020I \$A8: \$A7 \$0202 \$AA: \$A9 \$0203 \$AC: \$AB \$0204 \$AE:\$AD \$0205

\$00 R14 \$01 R15 \$A3 R16 \$17 R17

Idi ZL, \$00<<!
Idi ZH, \$02<<!
Ipm rI6, z+
Ipm rI7, z+</pre>

16 Bit

Z+I

\$01FF \$A2: \$A \$0200 **\$**A4:**\$**A3 \$A6: \$A5 \$020I \$A8: \$A7 \$0202 \$AA: \$A9 \$0203 \$AC: \$AB \$0204 \$AE:\$AD \$0205

\$00 R14 \$01 R15 \$A3 R16 \$17 R17

Idi ZL, \$00<<1
Idi ZH, \$02<<1
Ipm r16, z+
Ipm r17, z+

16 Bit

\$A2:\$AI

\$A4: **\$**A3

\$A6: \$A5

\$A8: \$A7

\$AA: \$A9

\$AC: \$AB

\$AE:\$AD

\$01FF

\$0200

\$020 I

\$0202

\$0203

\$0204

\$0205

8 Bit

\$00

\$01

\$A3

\$A4

R14

R15

R16

RI7

ldi ZL, \$00<<1

Idi ZH, \$02<< I

lpm r16, z+

lpm r17, z+

16 Bit

\$A2:\$AI

\$A4: \$A3

Z+ \$A5

\$A8: \$A7

\$AA: \$A9

\$AC: \$AB

\$AE:\$AD

\$01FF

\$0200

\$020I

\$0202

\$0203

\$0204

\$0205

8 Bit

\$00

\$01

\$A3

\$A4

R14

R15

R16

R₁7

ldi ZL, \$00<<1 ldi ZH, \$02<<1

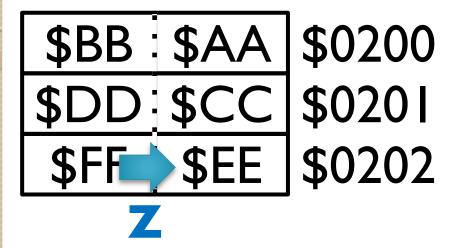
lpm r16, z+

lpm r17, z+

\$BB	\$AA	\$0200
\$DD	\$CC	\$0201
\$FF	\$EE	\$0202

\$16	RI6
\$17	RI7
\$18	RI8
\$19	RI9

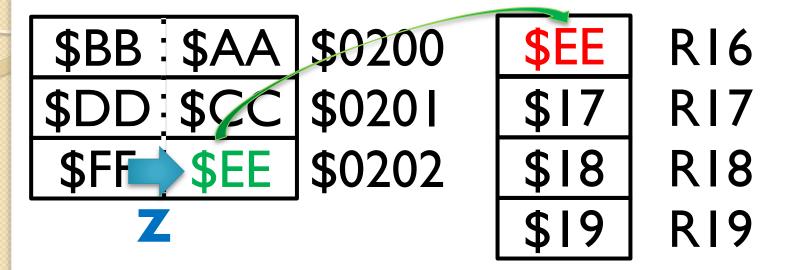
\$D1 \$0100 \$D2 \$0101 \$D3 \$0102 \$D4 \$0103



\$16	RI6
\$17	RI7
\$18	RI8
\$19	RI9

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

Idi ZL,\$02<<1 Idi ZH, \$02<<1 Ipm r16, Z



\$D1 \$0100 \$D2 \$0101 \$D3 \$0102 \$D4 \$0103 Idi ZL,\$02<<1 Idi ZH, \$02<<1 Ipm r16, Z

\$BB	\$AA	\$0200
\$DD	\$CC	\$020 I
\$FF \$EE \$0202		
7	7	•

\$EE	
\$17	
\$18	
\$19	

R16 R17 R18 R19



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

Idi ZL,\$02<<I Idi ZH, \$02<<I Ipm r16, Z Idi YL, \$00 Idi YH, \$01

\$BB \$AA	\$0200	\$EE	RI
\$DD:\$CC	\$020 I	\$17	RI
\$FF \$EE	\$0202	\$18	RI
Z		\$19	RI

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

Idi ZL,\$02<<!
Idi ZH, \$02<<!
Ipm r16, Z
Idi YL, \$00
Idi YH, \$01
st Y, r16</pre>

LCD Display

- Address of Data memory
 - Ist line of LCD: \$0100 \$010F
 - 2nd line of LCD: \$0110 \$011F

Stored Data in Program memory

```
    STRING I_BEG:

            .DB "Han Jang"

    STRING2_BEG:

            .DB "HelloWorld"
```

: denotes the name of address

Text Scroll from left to right

- Refer to Data Transfer instructions in Ch.4: Atmel's AVR 8-bit Microcontroller, Part 1
 - Assembly Programming

Check-off Lists

- Correct display functions accordingly with correction switch buttons.
- LCD Scrolling from left to right
- No dummy characters.
- Correct use of program memory.
- Correct use of data manipulation.

Questions?

