ECE375 Lab 4

TA: Youngbin Jin
School of Electrical Engineering and Computer Science
Oregon State University

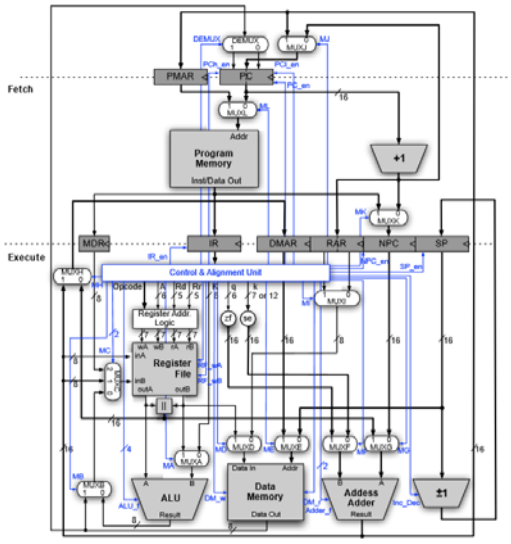
Demo and Plagiarism

- Show your demo.
  - Even your demo does not run successfully, show your code and demo in order to get partial credit.
- Do not copy other's code.
  - It is never okay to represent another person's work as your own.

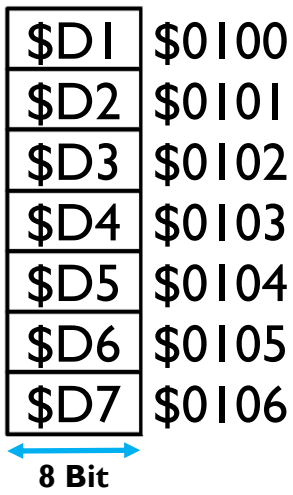
Data Manipulation & LCD

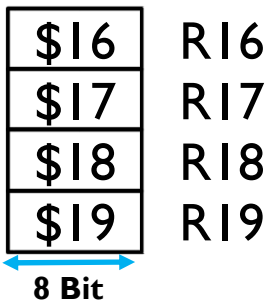
- Data Manipulation
  - Program memory <-> Register
  - Data memory <-> Register
  - Program memory <-> Data Memory
- Display LCD
  - LCD driver is provided

AVR Microarchitecture

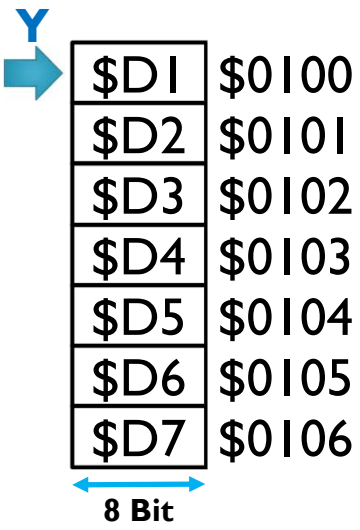


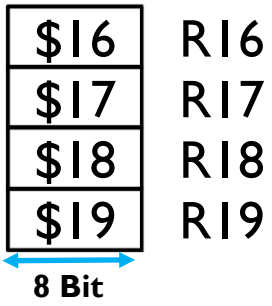
Data Memory to Register



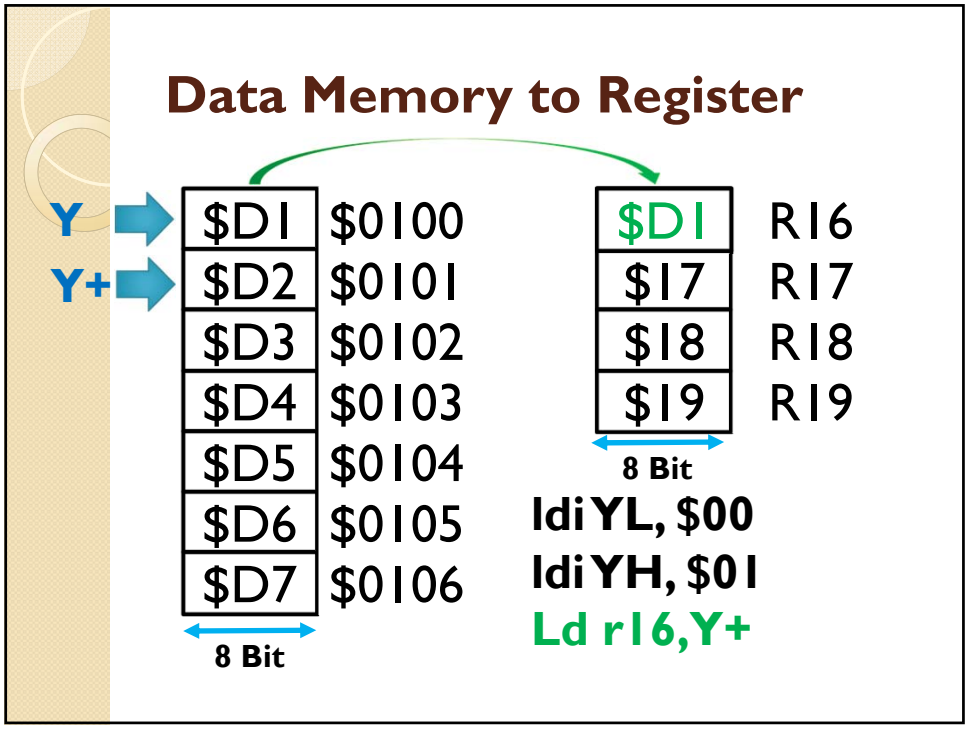


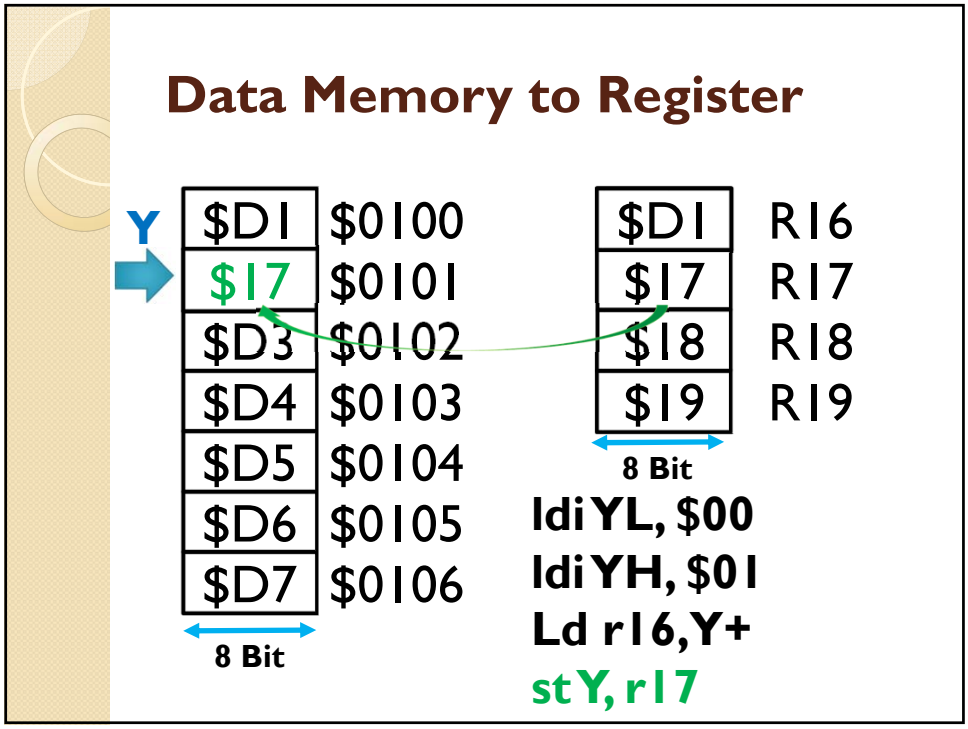
Data Memory to Register






```
ldi YL, $00
ldi YH, $01
```


Data Memory to Register
Y
Y+
$D1 $0100
$D2 $0101
$D3 $0102
$D4 $0103
$D5 $0104
$D6 $0105
$D7 $0106
8 Bit
$D1 R16
$17 R17
$18 R18
$19 R19
8 Bit
ldi YL, $00
ldi YH, $01
Ld r16,Y+


Data Memory to Register
Y
$D1 $0100
$17 $0101
$D3 $0102
$D4 $0103
$D5 $0104
$D6 $0105
$D7 $0106
8 Bit
$D1 R16
$17 R17
$18 R18
$19 R19
8 Bit
ldi YL, $00
ldi YH, $01
Ld r16,Y+
st Y, r17

Program Memory to Register

	16 Bit			8 Bit	
Z??	$A2	$A1	$0100	$00	R0
⇨	$A4	$A3	$0200	$01	R1
	$A6	$A5	$0201	$16	R16
	$A8	$A7	$0202	$17	R17
	$AA	$A9	$0203		
	$AC	$AB	$0204		
	$AE	$AD	$0205		

ldi ZL, $00
ldi ZH, $02

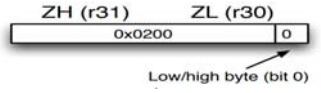
Program Memory(16 bits) ≠ Register (8bits)
Cannot move directly

Program Memory to Register

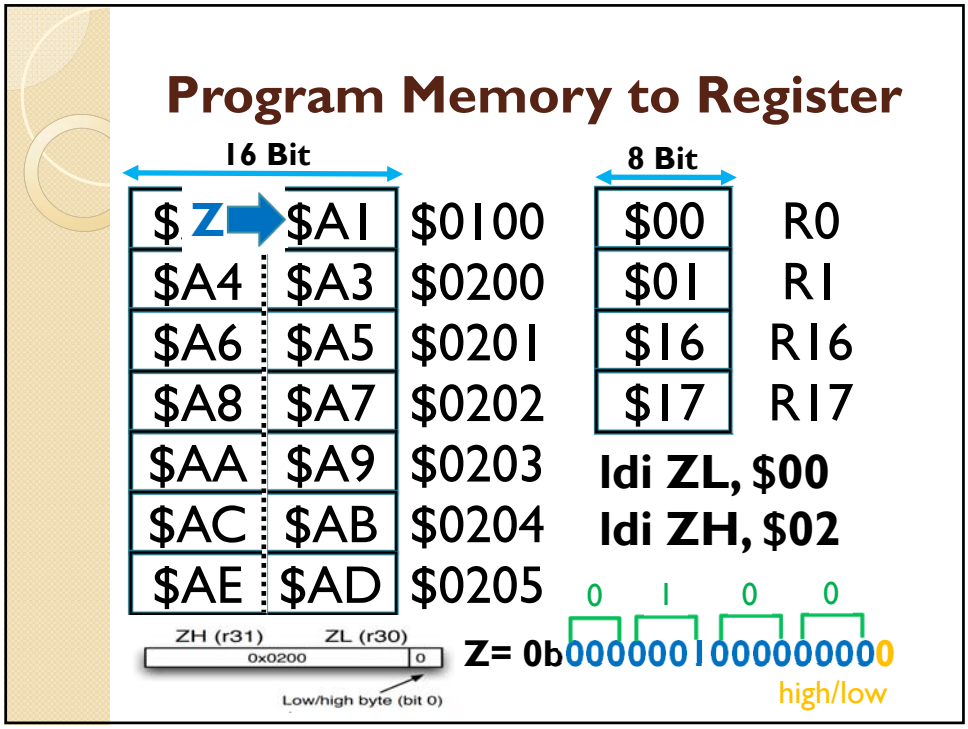
	16 Bit			8 Bit	
	$A2	$A1	$0100	$00	R0
	$A4	$A3	$0200	$01	R1
	$A6	$A5	$0201	$16	R16
	$A8	$A7	$0202	$17	R17
	$AA	$A9	$0203		
	$AC	$AB	$0204		
	$AE	$AD	$0205		

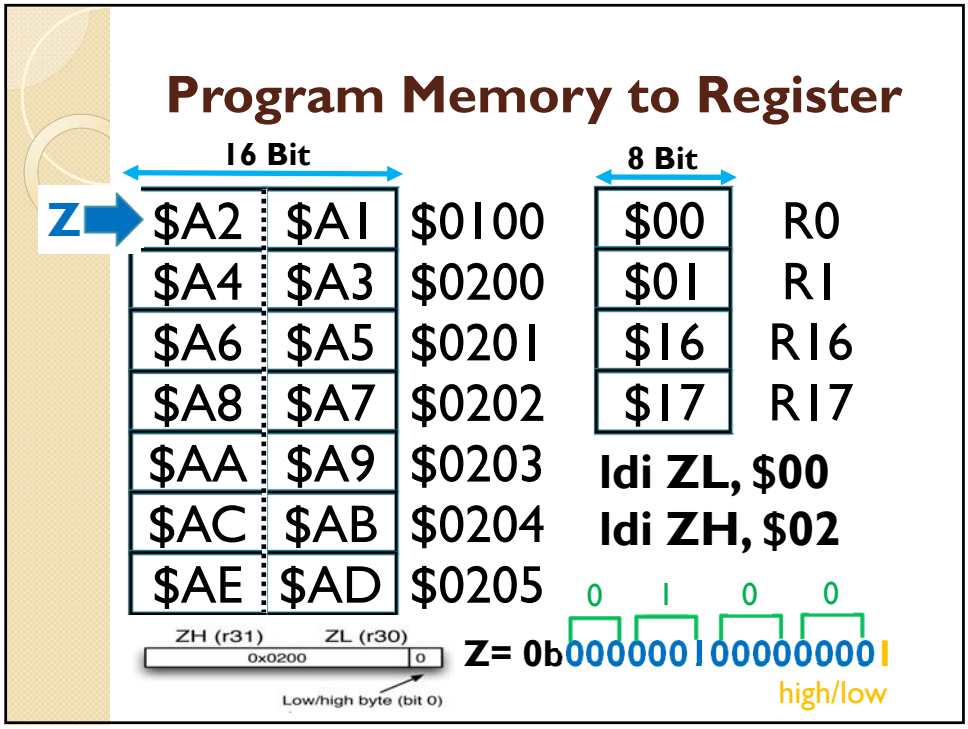
ldi ZL, $00
ldi ZH, $02

ZH 0 2 ZL 0 0

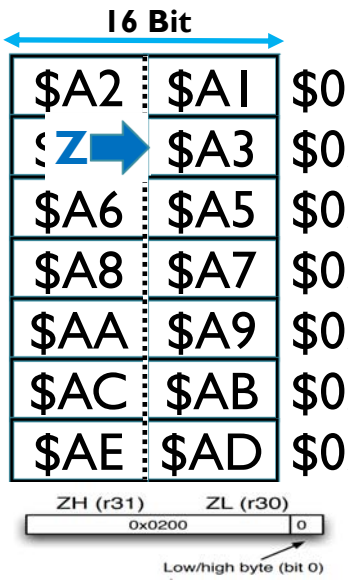


Z= 0b0000001000000000


Program Memory to Register
16 Bit
8 Bit
Z
$A1 $0100
$A4 $A3 $0200
$A6 $A5 $0201
$A8 $A7 $0202
$AA $A9 $0203
$AC $AB $0204
$AE $AD $0205
$00 R0
$01 R1
$16 R16
$17 R17
ldi ZL, $00
ldi ZH, $02
ZH (r31) ZL (r30)
0x0200 0
Low/high byte (bit 0)
0 1 0 0
Z= 0b0000001000000000
high/low


Program Memory to Register
16 Bit
8 Bit
Z
$A2 $A1 $0100
$A4 $A3 $0200
$A6 $A5 $0201
$A8 $A7 $0202
$AA $A9 $0203
$AC $AB $0204
$AE $AD $0205
$00 R0
$01 R1
$16 R16
$17 R17
ldi ZL, $00
ldi ZH, $02
ZH (r31) ZL (r30)
0x0200 0
Low/high byte (bit 0)
0 1 0 0
Z= 0b0000001000000001
high/low

Program Memory to Register



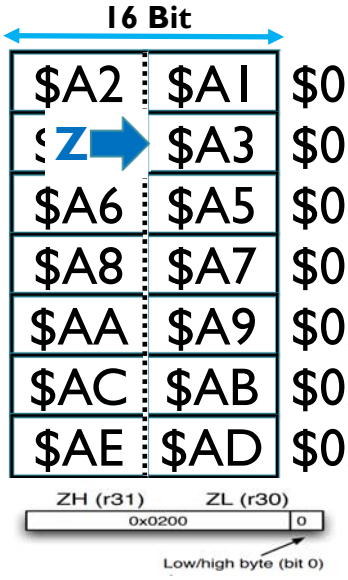

```
ldi ZL, $00
ldi ZH, $02
```

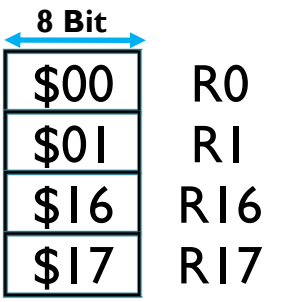
```
ldi ZL, ($00<<1)
ldi ZH, ($02<<1)
```

or

```
ldi ZL, $00
ldi ZH, $04
```

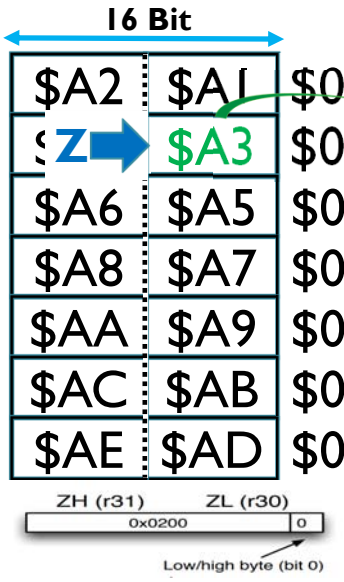
Program Memory to Register





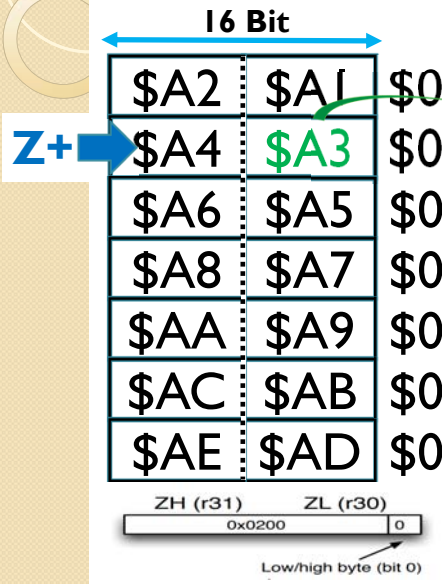

```
ldi ZL, $00 <<1
ldi ZH, $02 <<1
lpm r16, z+
lpm r17, z+
spm z+
```

Program Memory to Register



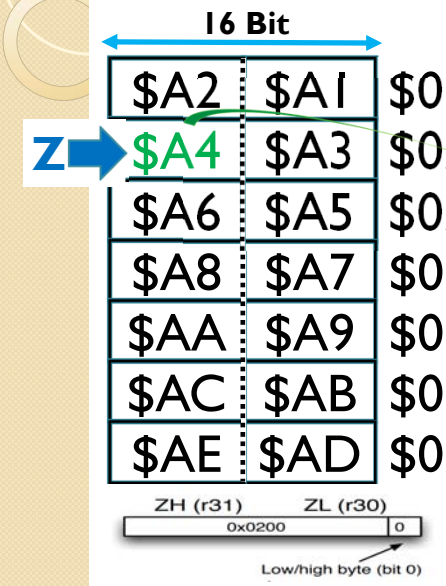

```
ldi ZL, $00<<1
ldi ZH, $02<<1
lpm r16, z+
lpm r17, z+
spm z+
```

Program Memory to Register



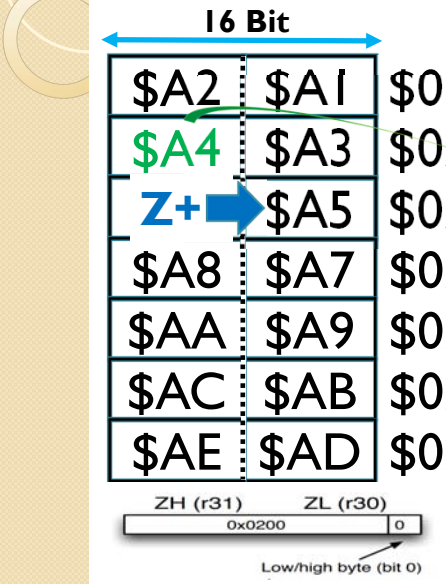

```
ldi ZL, $00<<1
ldi ZH, $02<<1
lpm r16, z+
lpm r17, z+
spm z+
```

Program Memory to Register



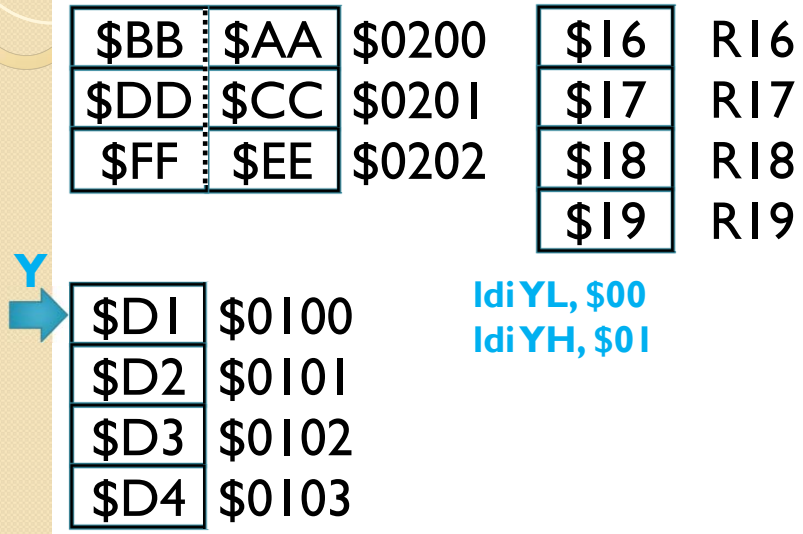

```
ldi ZL, $00<<1
ldi ZH, $02<<1
lpm r16, z+
lpm r17, z+
spm z+
```

Program Memory to Register

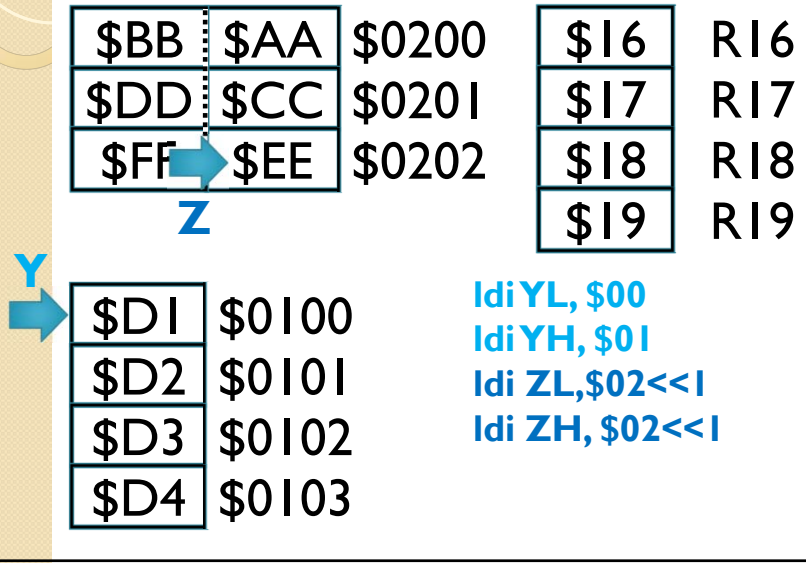


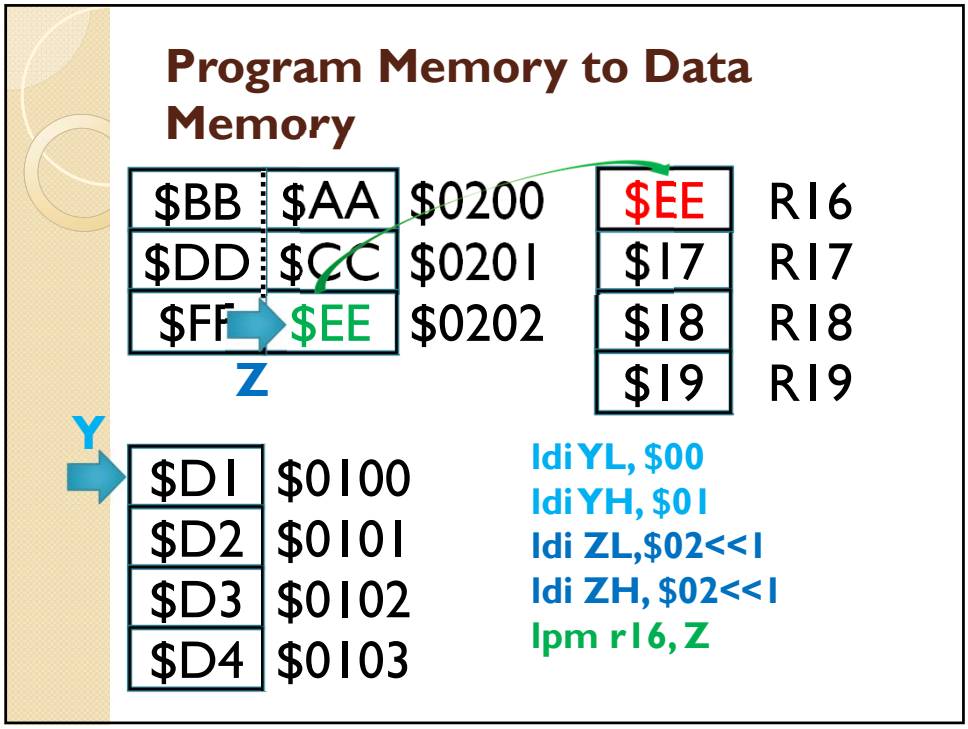

```
ldi ZL, $00<<1
ldi ZH, $02<<1
lpm r16, z+
lpm r17, z+
spm z+
```

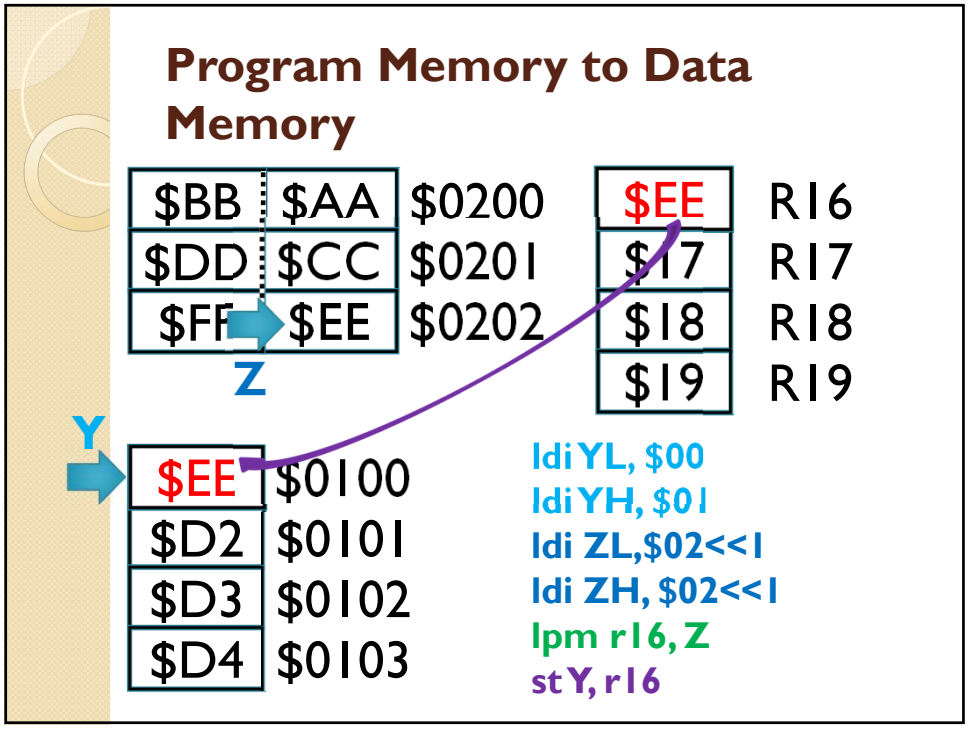
Program Memory to Data Memory



Program Memory to Data Memory




Program Memory to Data Memory
$BB $AA $0200
$DD $CC $0201
$FF $EE $0202
Z
$EE R16
$17 R17
$18 R18
$19 R19
Y
$D1 $0100
$D2 $0101
$D3 $0102
$D4 $0103
ldi YL, $00
ldi YH, $01
ldi ZL,$02<<1
ldi ZH, $02<<1
lpm r16, Z


Program Memory to Data Memory
$BB $AA $0200
$DD $CC $0201
$FF $EE $0202
Z
$EE R16
$17 R17
$18 R18
$19 R19
Y
$EE $0100
$D2 $0101
$D3 $0102
$D4 $0103
ldi YL, $00
ldi YH, $01
ldi ZL,$02<<1
ldi ZH, $02<<1
lpm r16, Z
st Y, r16

Display LCD

- LCD driver provided
  - Add LCD driver in your main asm file.
- LCDWrite
  - Display 1st line : $0100 - $010F
  - Display 2nd line : $0110 - $011F
- Store Data in Program Memory
  - DO NOT make redundant data
    - EX) .DB "Youngbin Jin Dongjun Lee "
      .DB "Dongjun Lee Youngbin Jin "
    - Use pointer properly
- Move to Data Memory $0100-$011F before rcall LCDWrite function

Demo

- Button 0 (PD0)
  - Youngbin Jin (1st String)
  - Dongjun Lee (2nd String)
- Button 1 (PD1)
  - Dongjun Lee (2nd String)
  - Youngbin Jin (1st String)
- Button 7 (PD7)
  - Clear

Checklist for Lab 4

- Demo Checklist
  - Strings displayed on both lines of LCD?
  - Strings are controlled by PD0, PD1, and PD7?
  - No garbage/uninitialized characters?
  - Strings declared in ProgMem using .DB?
  - Strings copied to DataMem using a loop?
  - Sufficient explanation of <<1 for LPM?
- Challenge Checklist
  - Strings scroll from line 1 to line 2?
  - Strings scroll can be controlled by PD5 and PD6?
  - Reasonable scrolling interval (~0.25 sec)?

Questions?