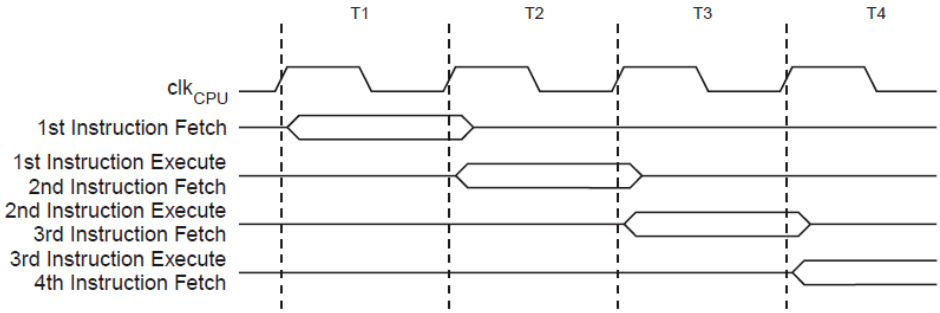
การทำงานและคำสั่งของ AVR Microcontrollers

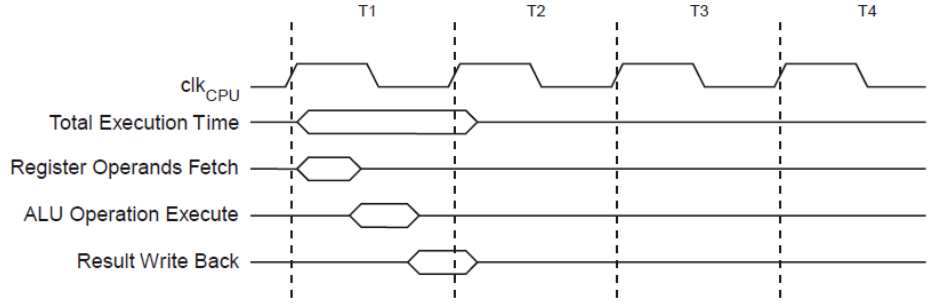
บทนำ

Instruction Execution Timing

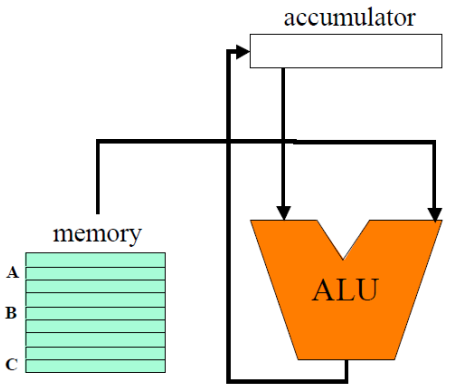
The Parallel Instruction Fetches and Instruction Executions



Single Cycle ALU Operation



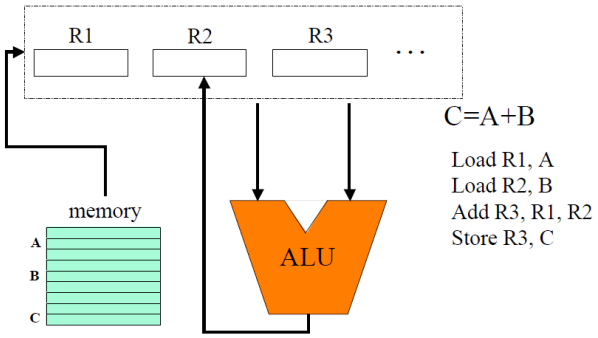
ลักษณะการทำงานของ CISC และ RISC processors



C=A+B

Load A
Add B
Store C

CISC processors have limited accumulators,
complex instructions,
orthogonal addressing modes.
Load data into accumulator.
Execute instructions with additional data from memory.
Store results into memory.



RISC processors have many working registers, simple addressing modes, memory access via load/store instructions.
Load data from memory into registers
Execute instructions using registers
Store results into memory

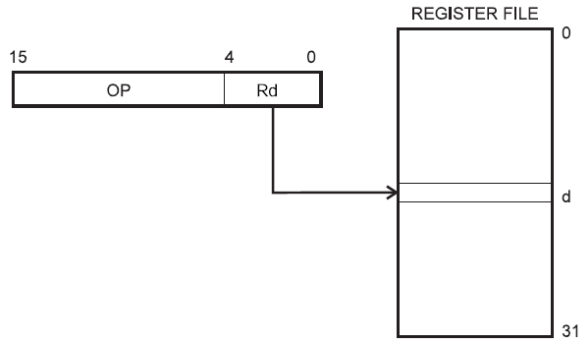
Addressing Modes

- **Register**: operand is in the register
- **Immediate**: operand is in the instruction itself
- **Direct**: address of operand is in the instruction
- **Register Direct**: address of operand is in the given register
- **Base Displacement**: address is the sum of register content and a constant
- **Indirect**: instruction have address, the contents at the address is the address of the operand
- **Register Indirect**: register contains address, the contents at the address is the address of the operand

- Addressing modes for code
  - Direct: jump addr
  - Register Direct: jump reg
  - Base Displacement: jump offset
- Addressing modes for data
  - Direct: load reg,addr
  - Base Displacement: load reg,base,offset

ตัวอย่าง Addressing Modes

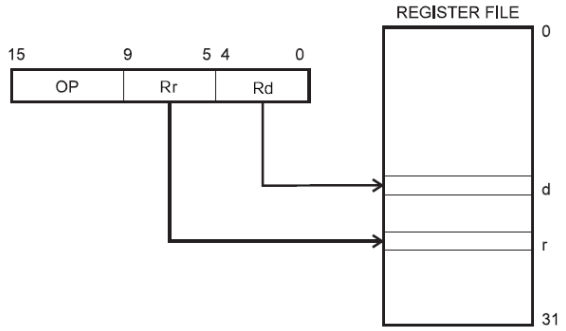
Register Direct, Single Register Rd



The operand is contained in register d (Rd).
เช่น

```
INC     R16
CLR     R22
EOR     R0
```

Register Direct, Two Registers Rd and Rr

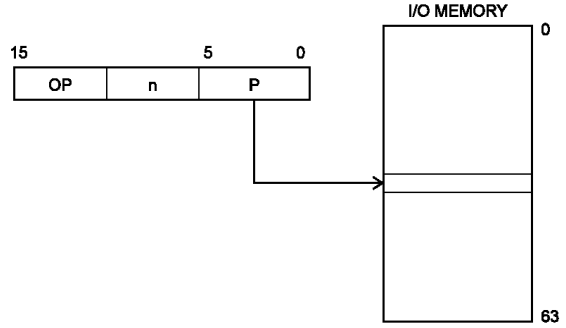


Operands are contained in register r (Rr) and d (Rd). The result is stored in register d (Rd).

เช่น

```
ADD     R16,R17
CP      R22,R5
MOV     R0, R1
```

IO Direct

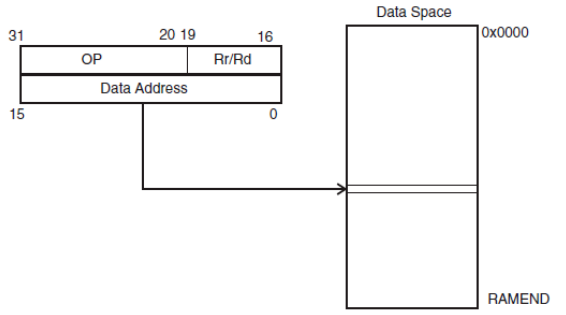


Operand address is contained in 6 bits of the instruction word. n is the destination or source register address.

เช่น

```
IN      R16,PIND
OUT     PORTC,R16
```

Data Direct

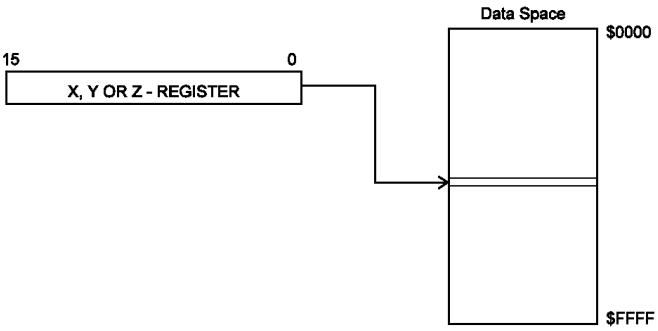


A 16-bit Data Address is contained in the 16 LSBs of a two-word instruction. Rd/Rr specify the destination or source register.

เช่น

```
STS     0x1000,R16
```

Data Indirect

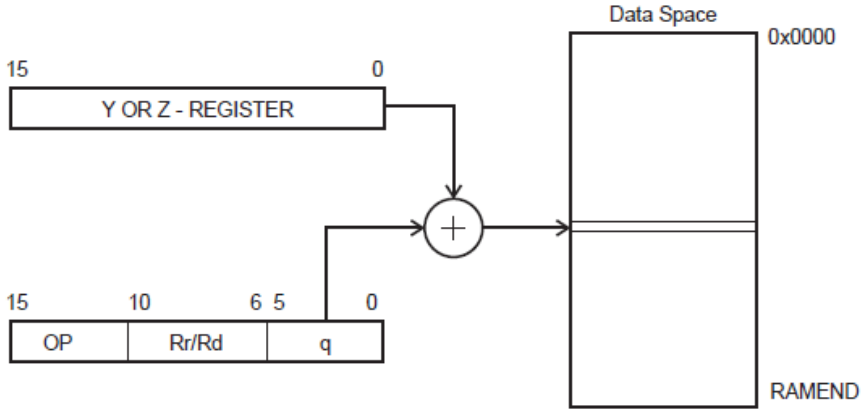


Operand address is the contents of the X-, Y-, or the Z-register. In AVR devices without SRAM, Data Indirect Addressing is called Register Indirect Addressing. Register Indirect Addressing is a subset of Data Indirect Addressing since the data space form 0 to 31 is the Register File.

เช่น

```
LD      R16, Y
ST      Z, R16
```

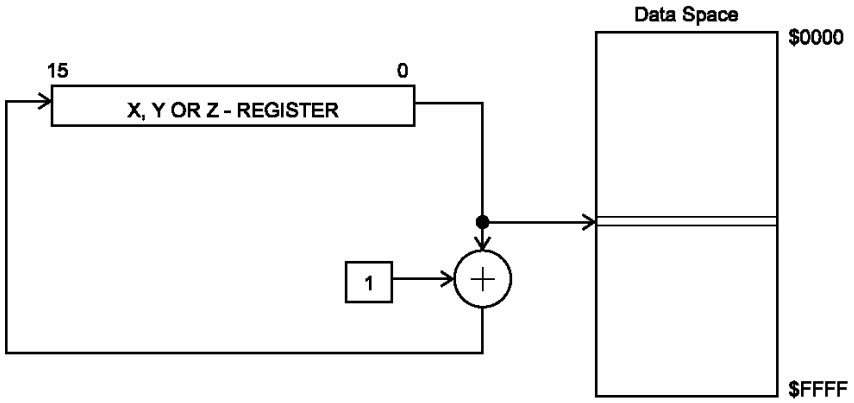
Data indirect with displacement



Operand address is the result of the Y- or Z-register contents added to the address contained in 6 bits of the instruction word. Rd/Rr specify the destination or source register.เช่น

```
LDD     R16, Y+0x10
STD     Z+0x20, R16
```

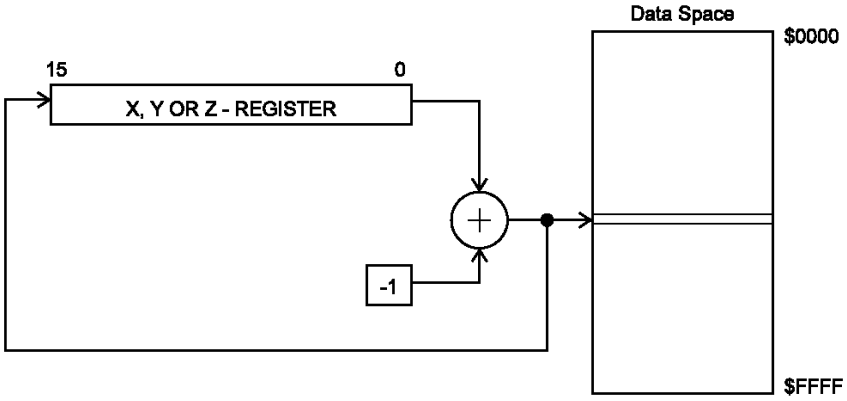
Data Indirect with Post-Increment



The X-, Y-, or the Z-register is incremented after the operation. Operand address is the content of the X-, Y-, or the Z-register prior to incrementing. เช่น

```
LD      R16, Z+
ST      Z+, R16
```

Data Indirect with Pre-Decrement



The X,- Y-, or the Z-register is decremented before the operation. Operand address is the decremented contents of the X-,Y-, or the Z-register.เช่น

```
LD      R16, -Z
ST      -Z, R16
```

Instruction Set

130 structions of ATmega8 are divided into following five groups:

- Arithmetic and logical instructions เช่น add subtract and or NOT
- Branch instructions
- Data transfer instructions
- Bit and bit-test instructions, and
- MCU control instructions

ตัวอย่างคำสั่ง : **ADD**

Math Add

ADD Rd, Rr – Adds two registers
Rd <- Rd + Rr

ADC Rd, Rr – Add with Carry two registers
Rd <- Rd + Rr + C

ADIW Rdl, K – Add Immediate to Word
Rdh:Rdl <- Rdh:Rdl + K

คำสั่งทางคณิตศาสตร์และโลจิกแบบอื่น ๆ Subtract • Logical AND • Logical OR • Exclusive OR • One's Complement • Two's Complement • Increment/Decrement • Set/Clear Registers and Bits in Registers	**คำสั่งเกี่ยวกับการกระโดด** (**Branch instructions)** • RJMP/RCALL – Relative Jmp (+/-k) • IJMP/ICALL – Indirect Jmp (Z Reg) • RET/RETI – Return from call/interrupt • CP* - Compare • SB* - Skip if Bit in Register or I/O is set/clr • BR* - Branch if condition is met
คำสั่งเกี่ยวกับการเคลื่อนย้ายข้อมูล (**Data Transfer)** • MOV – Move between registers • LD/LDI – Load / Load Immediate • ST/STI – Store / Store Immediate • LPM – Load Program Memory – Hardwired to load R0 with (Z) in code. • IN/OUT – In and Out Ports • PUSH/POP – On and off stack	**คำสั่งเกี่ยวกับการตรวจสอบบิต** (**Bit and Bit test)** • SBI/CBI – Set / Clear Bit in register • LSL/LSR – Logical Shift Left / Right • ROL/ROR – Rotate Left / Right (thru Carry bit) • ASR – Arithmetic Shift Right • SWAP – Swap Nibbles • BST/BLD – Bit Store / Load • BSET/BCLR – Set / Clear Status Bits by number • SE*/CL* - Set / Clear Status Bits by name
คำสั่งอื่น ๆ • NOP – Do nothing for 1 cycle • SLEEP – Sleep until reset or interrupted • WDR – Watch Dog Reset	