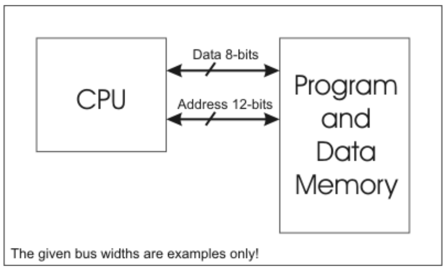
AVR Microcontrollers Architecture

บทนำ

สถาปัตยกรรมในการเข้าถึงหน่วยความจำของซีพียู หลักๆมี 2 แบบ

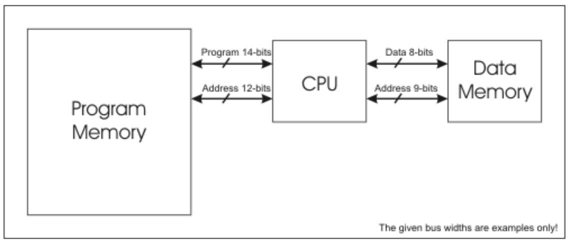
1. Von Neumann Architecture
2. Harvard Architecture

Von Neumann Architecture



John Von Neumann's: One shared memory for instructions (program) and data with one data bus and one address bus between processor and memory. Instructions and data have to be fetched in sequential order (known as the Von Neuman Bottleneck), limiting the operation bandwidth. Its design is simpler than that of the Harvard architecture. It is mostly used to interface to external memory.

Harvard Architecture



The term originated from the Harvard Mark 1 relay-based computer, which stored instructions on punched tape and data in relay latches.

Harvard Architecture: The Harvard architecture uses physically separate memories for their instructions and data, requiring dedicated buses for each of them. Instructions and operands can therefore be fetched simultaneously.

Different program and data bus widths are possible, allowing program and data memory to be better optimized to the architectural requirements. E.g.: If the instruction format requires 14 bits then program bus and memory can be made 14-bit wide, while the data bus and data memory remain 8-bit wide

AVR Microcontrollers

ลักษณะโดยทั่วไป

พัฒนาโดย Atmel ประมาณปี 1996

ใช้ Flash memory สำหรับเก็บโปรแกรม

เป็น RISC Microcontrollers ซึ่งใช้สถาปัตยกรรมแบบ Harvard

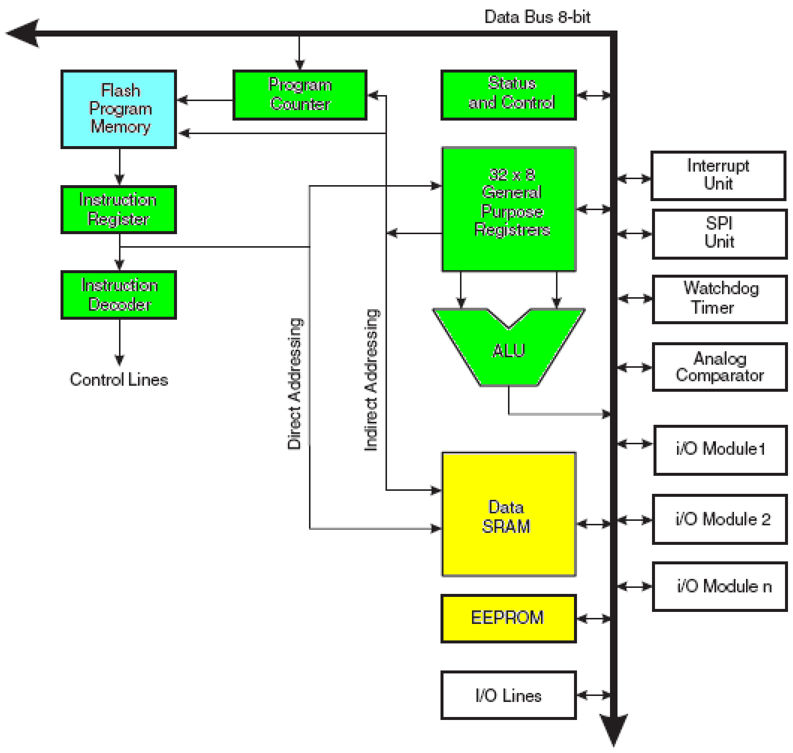
มีหลายรุ่นเช่น

- tiny AVR
- mega AVR
- XMEGA AVR
- 32-bit AVR

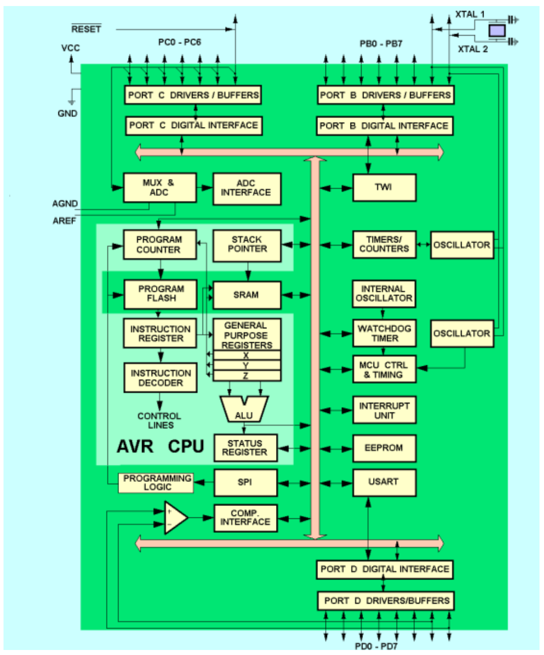
ลักษณะของ AVR อนุกรมต่างๆ

Series Name	Pins	Flash Memory	Special Feature
TinyAVR	6-32	0.5-8 KB	Small in size
MegaAVR	28-100	4-256KB	Extended peripherals
XmegaAVR	44-100	16-384KB	DMA , Event System included

Block diagram of the AVR MCU Architecture



Simplified Internal Architecture of ATmega8

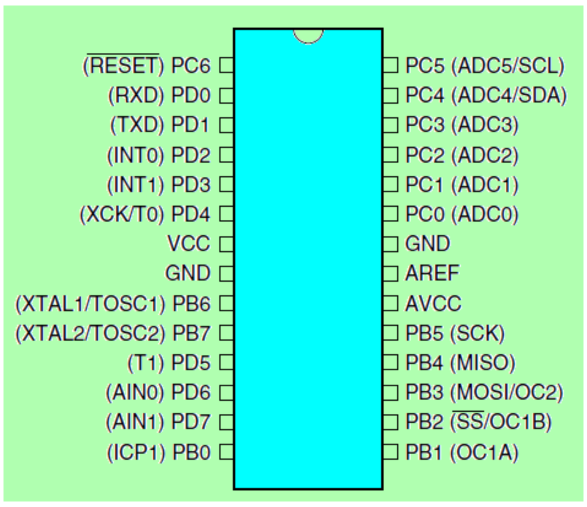


The central feature is the AVR core consisting ALU, 32 general purpose registers including three index registers, processor status register, stack pointer and program counter with instruction register and instruction decoder.

Some Important Features of ATmega8

- 8-bit advanced RISC architecture.
- Register to register architecture without any accumulator.
- 130 instructions including signed and unsigned multiplication.
- 32 general purpose 8-bit registers.
- Fully static to 16 MHz operational frequency.
- 8K bytes on-chip flash program memory.
- 1K bytes on-chip SRAM for data storage.
- 512 bytes on-chip EEPROM.
- Programming lock for software security.
- 23 programmable I/O lines.
- Two 8-bit and one 16-bit Timers/Counters.
- Six-channel 10-bit on-chip ADC.
- Programmable serial USART.
- SPI serial interface.
- Watchdog timer with separate on-chip oscillator.
- On-chip analog comparator.
- External and internal interrupt sources.
- Five sleep modes for power management.

Pins and Signals of ATmega8



- Note that all 23 I/O lines have alternate functions.
- Five power input lines.
- External crystal and reset are not essential.

Operating Voltage and Power Consumption

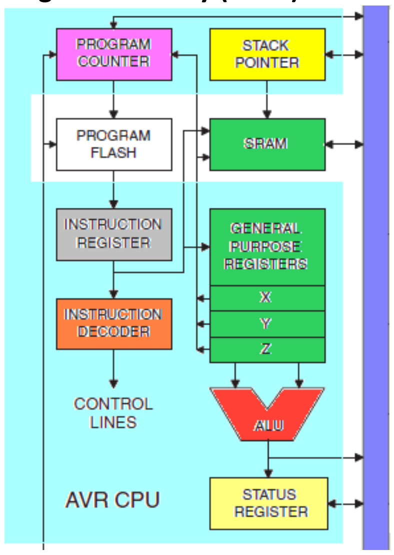
- Operating voltage of
- ATmega8: 4.5V to 5.5V
- ATmega8L: 2.7V to 5.5V

- At 4 Mhz with 3 volts it consumes 3.6 mA, reduced to 1 mA in idle mode and 0.5 uA in power-down mode.

- Each port pin of ATmega8L is capable of sourcing or sinking 20 mA current at 5V and 10 mA current at 3V.

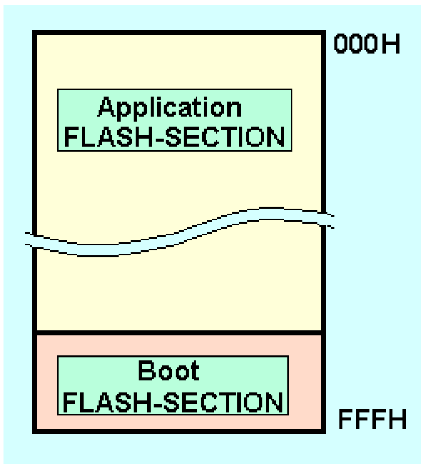
Note the identical sourcing as well as sinking capabilities.

Program Memory (Flash)



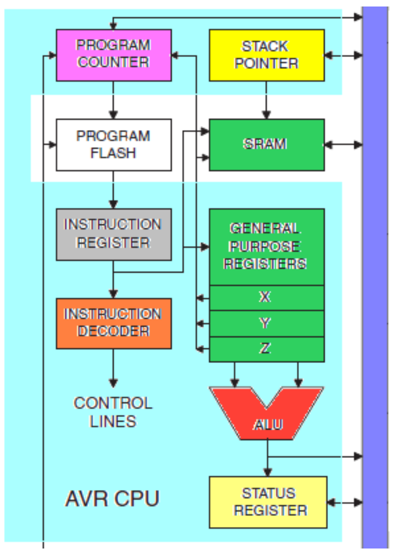
- The Flash memory has an endurance of at least 10,000 write/erase cycles.
- The ATmega8A Program Counter (PC) is 12 bits wide, thus addressing the 4K Program memory locations
- ถ้า Flash ขนาดใหญ่กว่านี้ PC ก็มากขึ้น

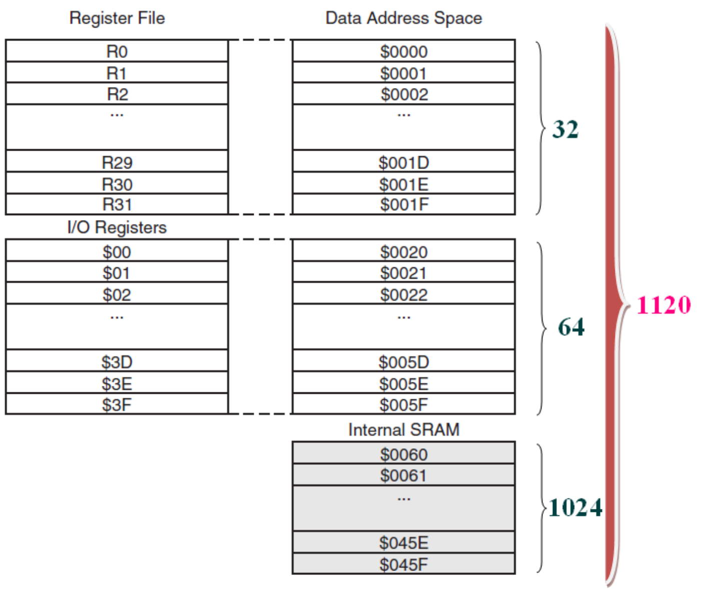
Map of ATmega8 Program Memory (Flash)



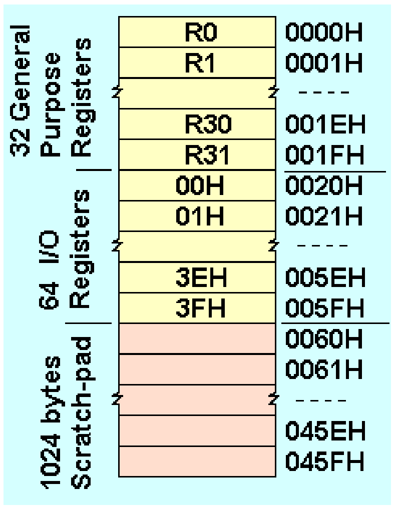
- Program memory is arranged as 4K words as ATmega8 opcodes are either 16-bit or 32-bit.
- The program memory is divided into two sections : boot and application for software security.

SRAM Data Memory and Register



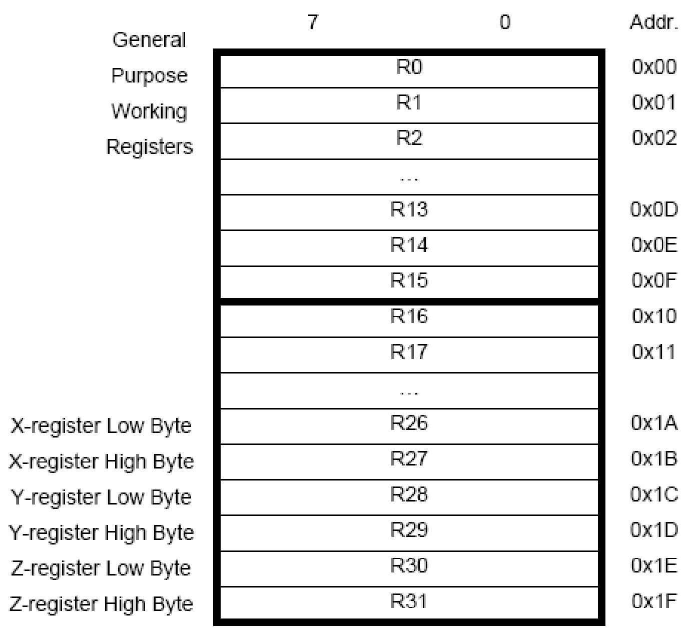


Organization of 1120 bytes of SRAM Data Memory



- Data memory (SRAM) accommodates general purpose registers, I/O registers and scratch pad area.
- I/O registers are similar to the Special Function Registers (SFRs) of 8051.
- I/O registers have two sets of addresses.

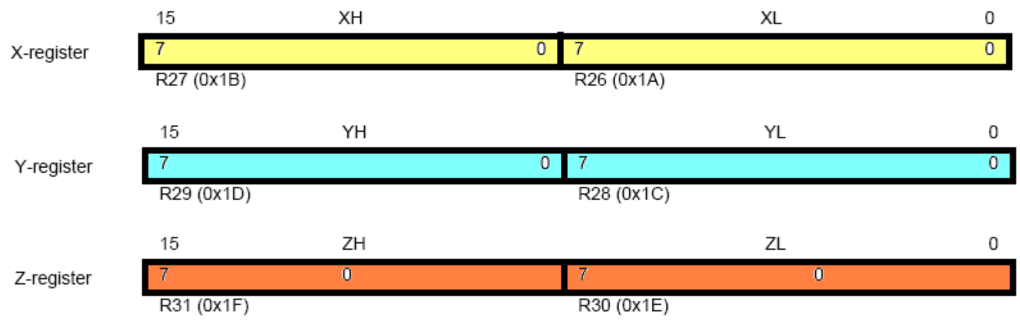
AVR CPU General Purpose Working Registers


General Purpose Working Registers
7
0
Addr.
R0 0x00
R1 0x01
R2 0x02
...
R13 0x0D
R14 0x0E
R15 0x0F
R16 0x10
R17 0x11
...
X-register Low Byte R26 0x1A
X-register High Byte R27 0x1B
Y-register Low Byte R28 0x1C
Y-register High Byte R29 0x1D
Z-register Low Byte R30 0x1E
Z-register High Byte R31 0x1F

- รีจีสเตอร์ทั้งหมดสามารถใช้ชุดคำสั่งเพื่อเข้าถึงได้โดยตรงและจะใช้ช่วงเวลาการเข้าถึงเพียง 1 Clock โดยคำสั่ง SBCI , SUBI , CPI , ANDI และ ORI ซึ่งกระทำ ระหว่างรีจีสเตอร์กับค่าคงที่และรีจีสเตอร์กับรีจีสเตอร์
- คำสั่ง LDI ที่ใช้โหลดค่าคงที่เข้าในรีจีสเตอร์ จะต้องใช้งานกับรีจีสเตอร์ R6 - R31
- ส่วนคำสั่ง SBC ,SUB,CP,AND และ OR และคำสั่งใช้งานอื่นๆสามารถใช้งานได้ในรีจีสเตอร์ทั่วไป

The X-register, Y-register and Z-register

รีจีสเตอร์ R26 - R31สามารถนำมาต่อกันเพื่อทำเป็นรีจีสเตอร์คู่เพื่อใช้งานเป็นตัวชี้ข้อมูลแบบ indirect addressing ในชื่อของ รีจีสเตอร์ X , Y และ Z



Details of Status Register (SREG) of ATmega8

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7 – I: Global Interrupt Enable
The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

• Bit 6 – T: Bit Copy Storage
The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

• Bit 5 – H: Half Carry Flag
The Half Carry Flag H indicates a half carry in some arithmetic operations. Half Carry is useful in BCD arithmetic.

• Bit 4 – S: Sign Bit, $S = N \oplus V$
The S-bit is always an exclusive or between the Negative Flag N and the Two's Complement Overflow Flag V.

• Bit 3 – V: Two's Complement Overflow Flag
The Two's Complement Overflow Flag V supports two's complement arithmetics.

• Bit 2 – N: Negative Flag
The Negative Flag N indicates a negative result in an arithmetic or logic operation.

• Bit 1 – Z: Zero Flag
The Zero Flag Z indicates a zero result in an arithmetic or logic operation.

• Bit 0 – C: Carry Flag
The Carry Flag C indicates a carry in an arithmetic or logic operation.

Stack and Stack Pointer
Stack may be located anywhere within upper 1024 bytes of SRAM. Lower 96 bytes (register area) must not be occupied by stack.
16-bit stack pointer contains two 8-bit registers, SPH and SPL.
PUSH command decreases the stack pointer and POP command increases it.
<u>In general, the stack pointer to be initialized by the highest address of SRAM.</u>

SPH and SPL – Stack Pointer High and Low Register

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Stack Pointer Instructions

Instruction	Stack pointer	Description
PUSH	Decremented by 1	Data is pushed onto the stack
CALL ICALL RCALL	Decremented by 2	Return address is pushed onto the stack with a subroutine call or interrupt
POP	Incremented by 1	Data is popped from the stack
RET RETI	Incremented by 2	Return address is popped from the stack with return from subroutine or return from interrupt

Power Management and Sleep Modes

- ATmega8 offers five sleep modes for efficient power management. They are:
  - Idle mode
  - ADC noise reduction mode
  - Power-down mode
  - Power-save mode, and
  - Standby mode.
- SLEEP instruction evokes the sleep mode.

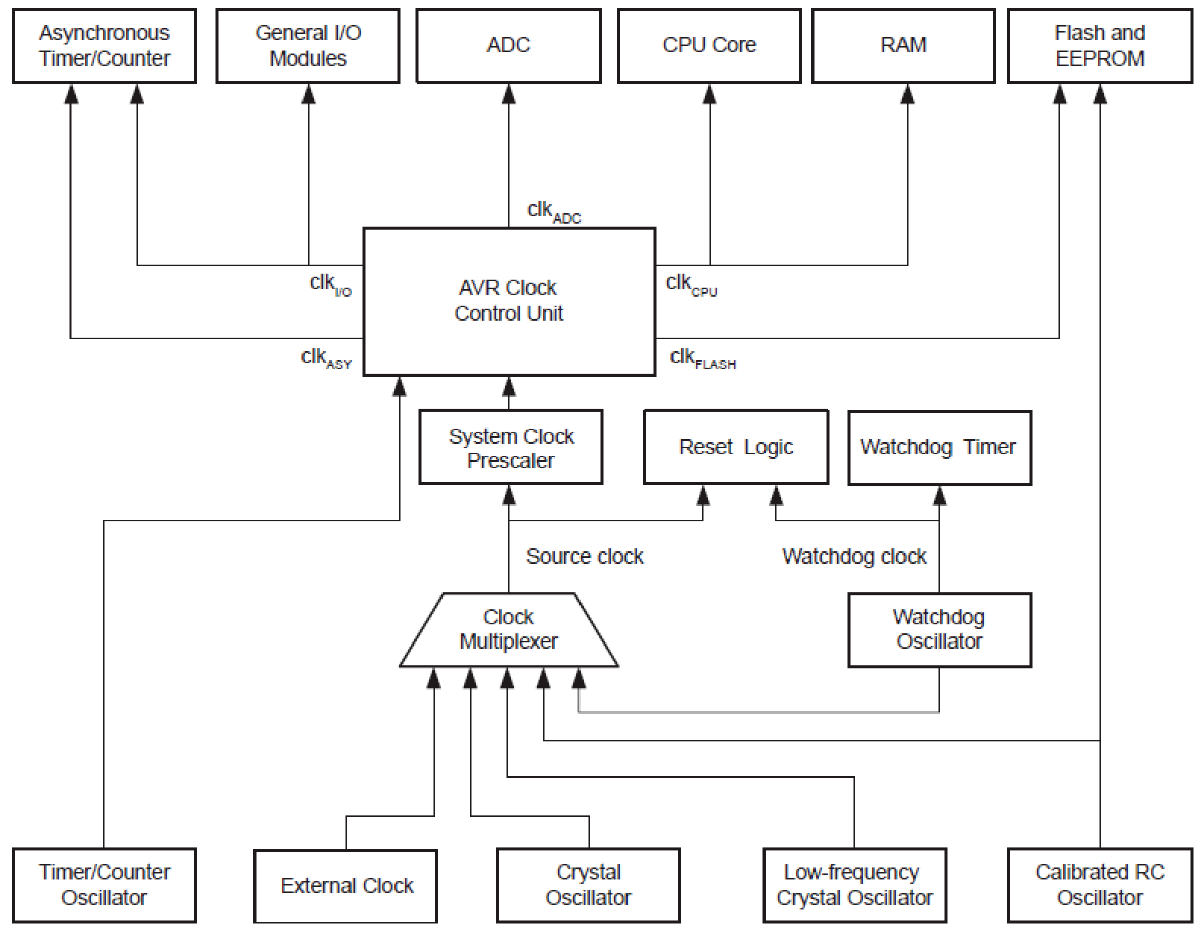
System Reset

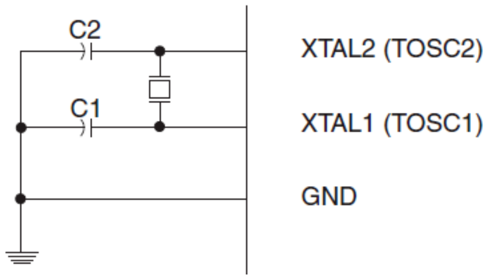
- Four sources of reset for ATmega8 are:
  - Power-on reset
  - External reset
  - Watchdog reset, and
  - Brown-out reset.
- All I/O registers are initialized and execution starts from the reset vector.
- MCUCSR register holds the information about the source of reset.

Watchdog Timer

- Operated by a separate internal oscillator of 1 MHz frequency.
- May be enabled or disabled through its register WDTCR.
- Generates system reset at terminal count.
- Useful to avoid 'hanging' system.
- To be loaded periodically to avoid system reset.

System Clock and Clock Options





- Clock source can be selected via I/O register
- Changing clock source requires time to stabilize frequency
- Adjusting clock frequency makes trade-off between processing power and power consumption