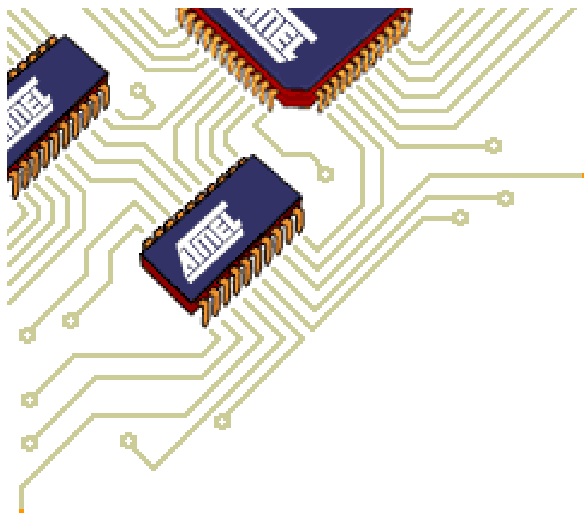
Microprocessor Development Tools



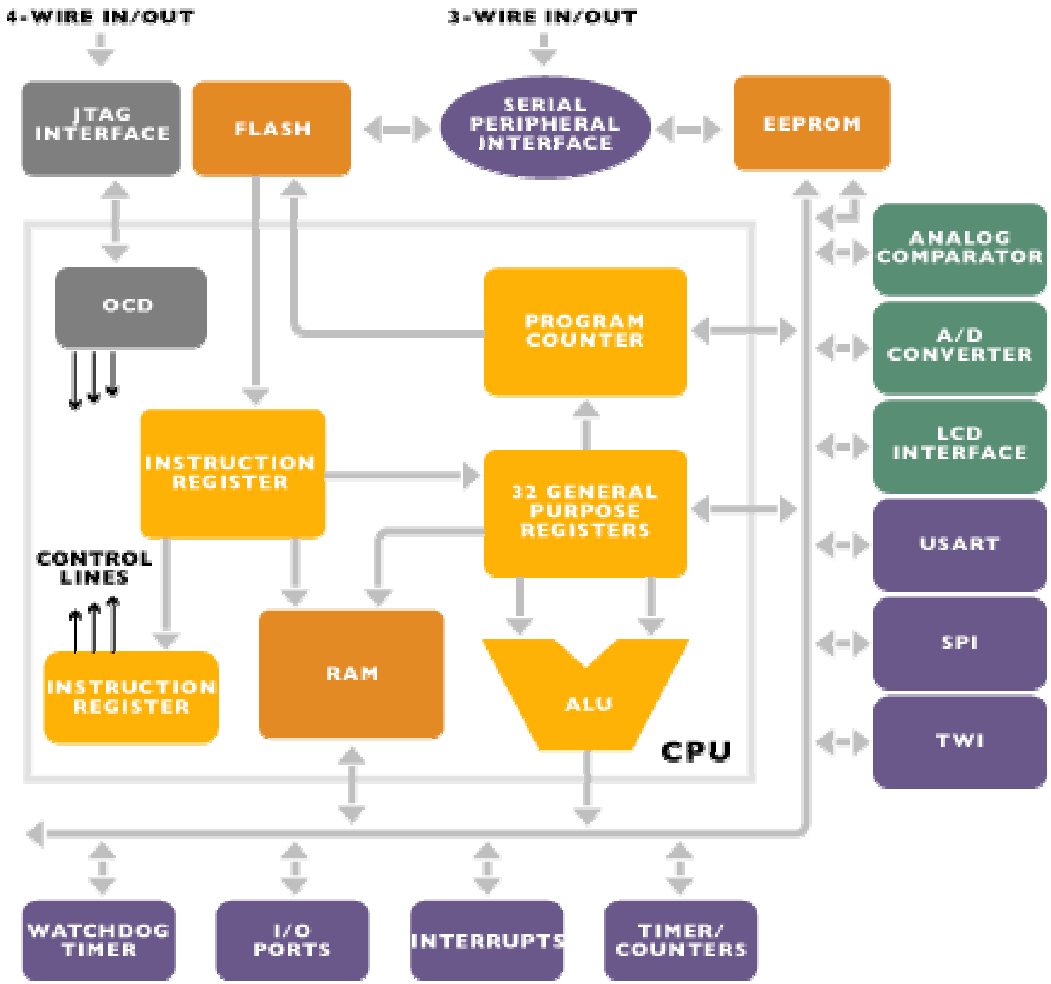
Tool Types

- **Software Tools**
  - **Assembler**
  - **Linker**
  - **Loader**
  - **Compiler**
  - **Libraries**
  - **Pre-compiled IP**
  - **Simulator**
  - **IDE**

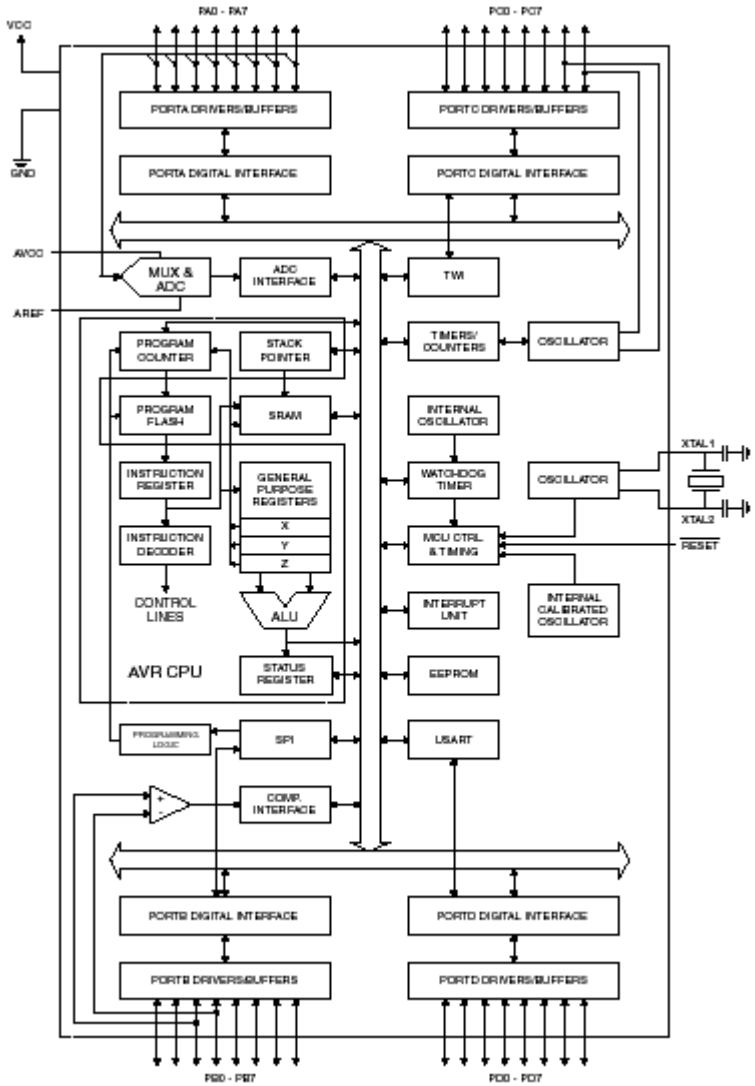
Hardware Tools

- **In Circuit Emulator (ICE)**
- **Logic Analyzer**
- **Microprocessor Development Systems**
- **System Development Systems**

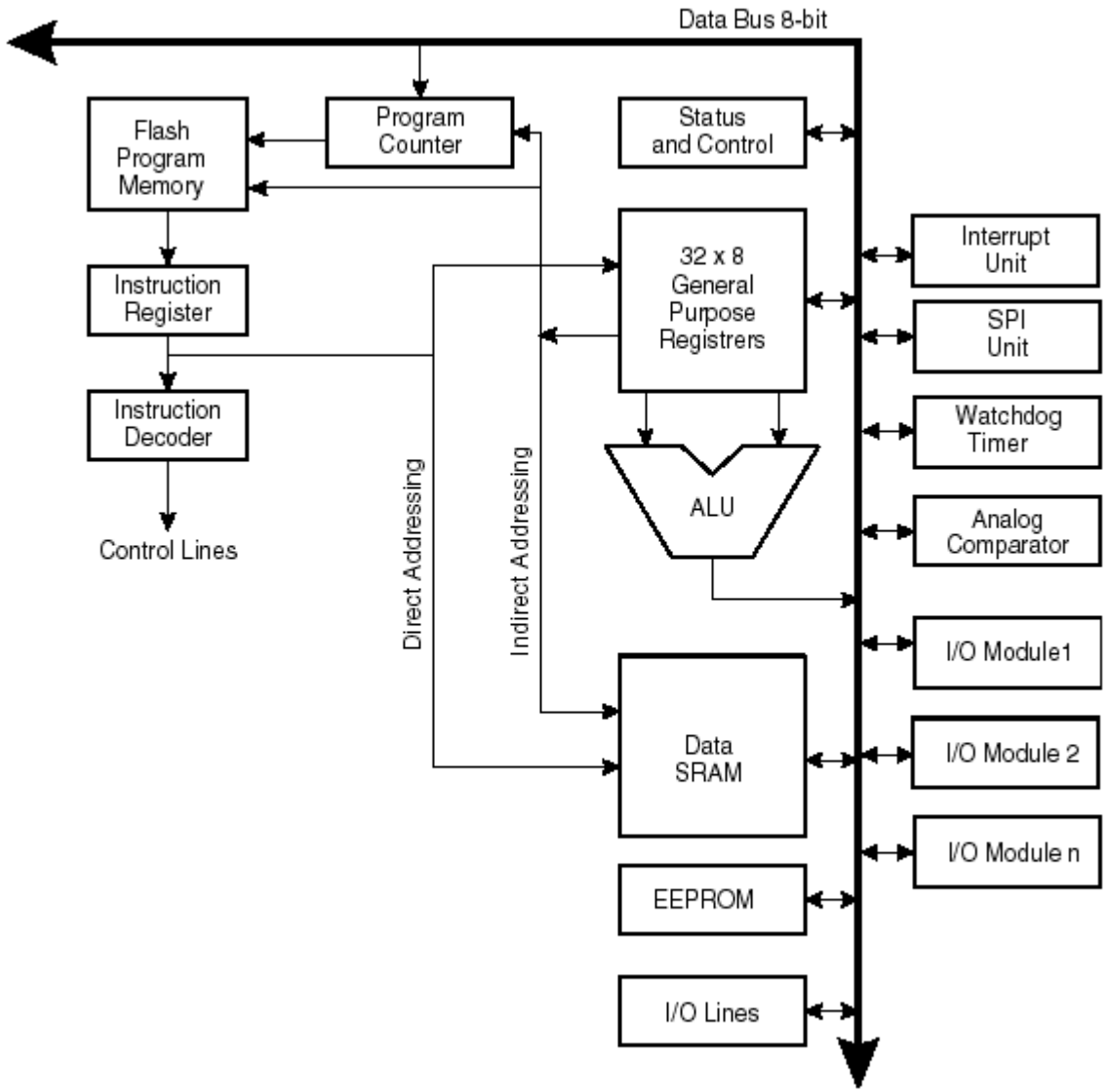
Atmel AVR Architecture



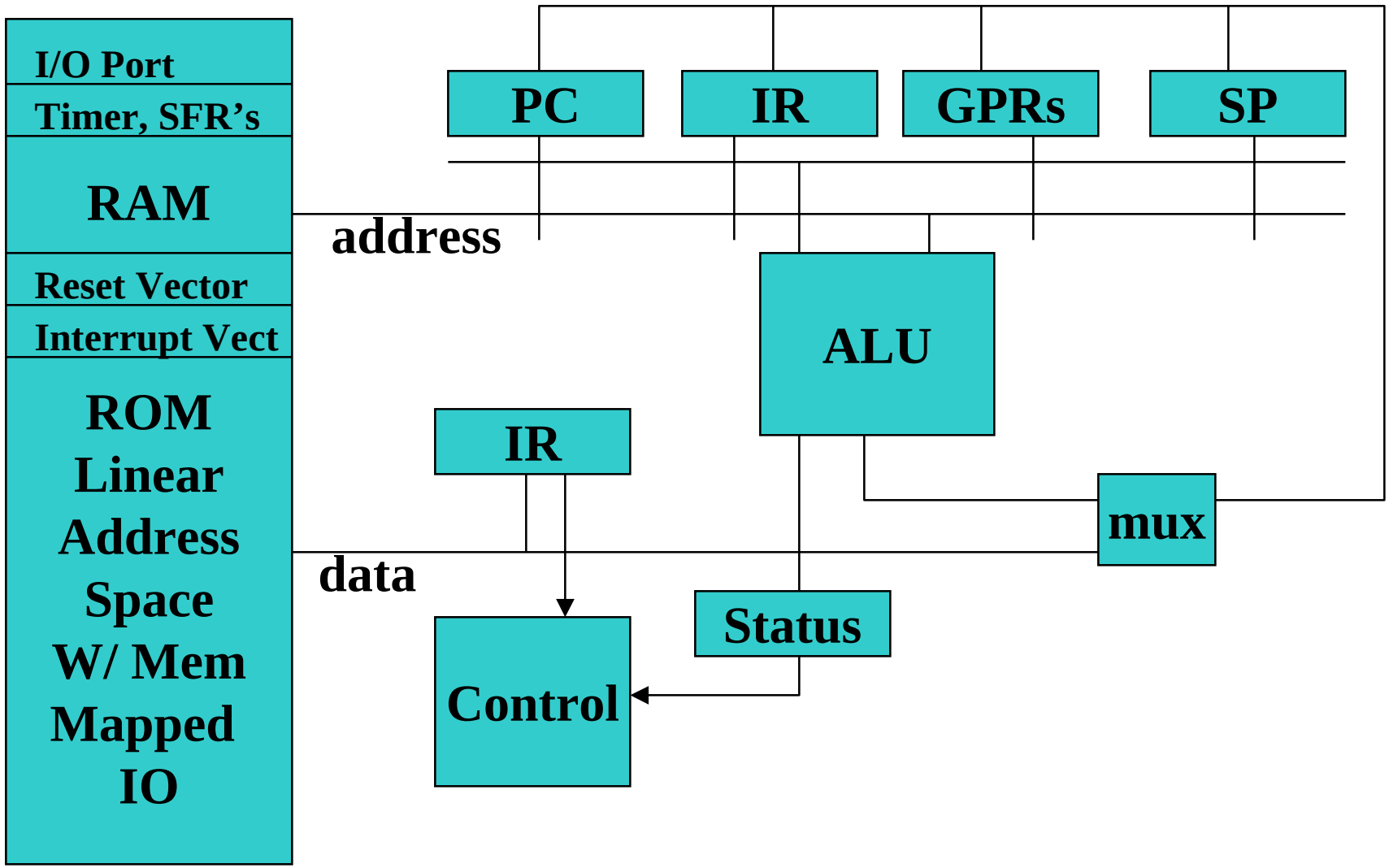
ATmega16


VCC
GND
AVCC
AREF
PA0 - PA7
PC0 - PC7
PORTA DRIVERS/BUFFERS
PORTC DRIVERS/BUFFERS
PORTA DIGITAL INTERFACE
PORTC DIGITAL INTERFACE
MUX & ADC
ADC INTERFACE
TWI
PROGRAM COUNTER
STACK POINTER
TIMERS/ COUNTERS
OSCILLATOR
PROGRAM FLASH
SRAM
INTERNAL OSCILLATOR
INSTRUCTION REGISTER
GENERAL PURPOSE REGISTERS
WATCHDOG TIMER
OSCILLATOR
XTAL1
XTAL2
INSTRUCTION DECODER
X
Y
Z
MCU CTRL & TIMING
RESET
CONTROL LINES
ALU
INTERRUPT UNIT
INTERNAL CALIBRATED OSCILLATOR
AVR CPU
STATUS REGISTER
EEPROM
SPI
USART
COMP. INTERFACE
PORTB DIGITAL INTERFACE
PORTD DIGITAL INTERFACE
PORTB DRIVERS/BUFFERS
PORTD DRIVERS/BUFFERS
PB0 - PB7
PD0 - PD7

AVR CPU


Data Bus 8-bit
Flash Program Memory
Program Counter
Status and Control
Interrupt Unit
SPI Unit
Watchdog Timer
Analog Comparator
I/O Module1
I/O Module 2
I/O Module n
Instruction Register
Instruction Decoder
Control Lines
32 x 8 General Purpose Registrers
ALU
Direct Addressing
Indirect Addressing
Data SRAM
EEPROM
I/O Lines

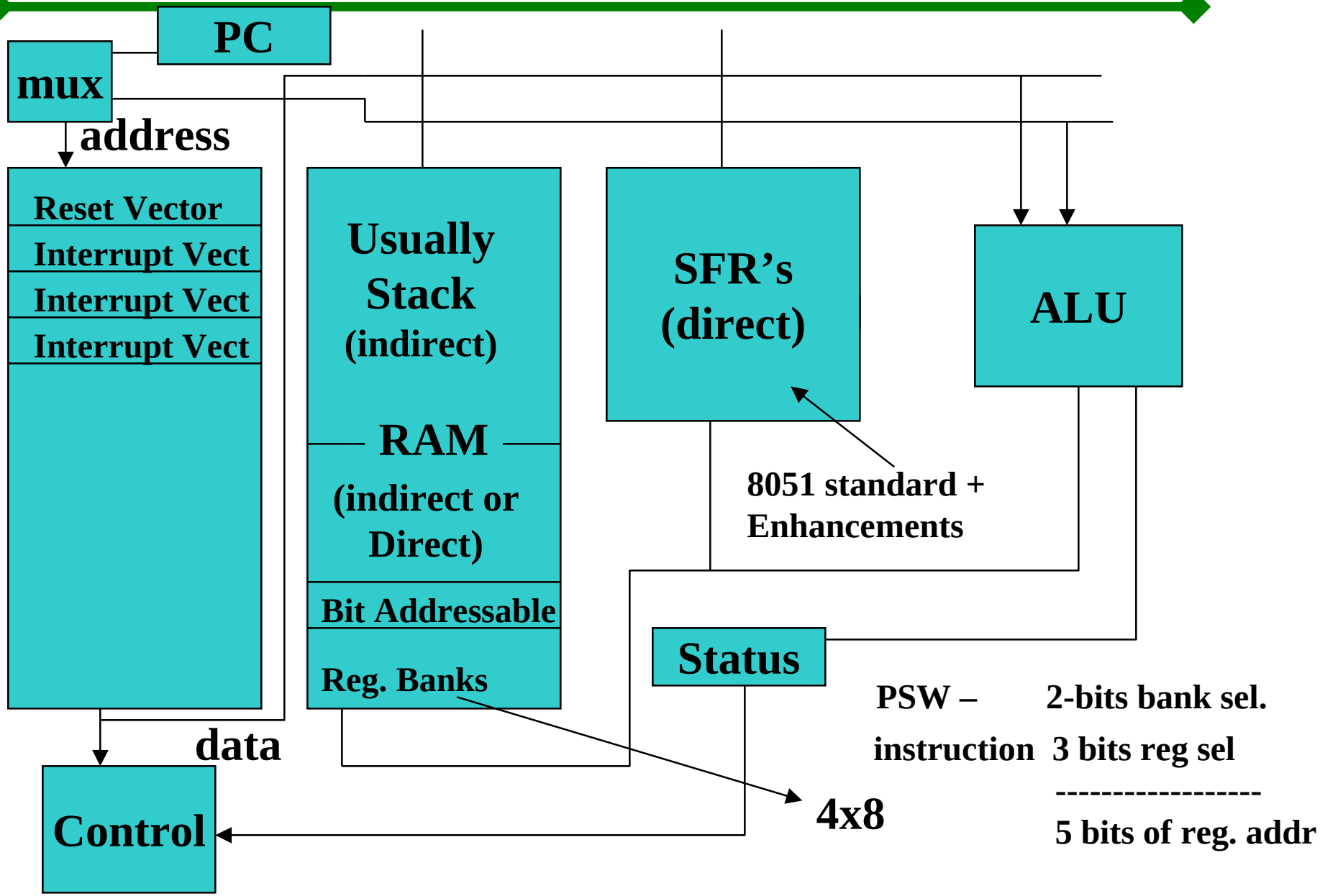
Simple Princeton Architecture



Analysis

- **Bottleneck into and out of memory for data and code**
- **Use of critical 8-bit address space (256) for memory mapped I/O and special function registers (timers and their controllers, interrupt controllers, serial port buffers, stack pointers, PC, etc). For example, the Motorola 6805 processor has only 187 RAM locations.**
- **But, easy to program and debug. Compiler is simple too.**

8051: Modified Harvard Architecture

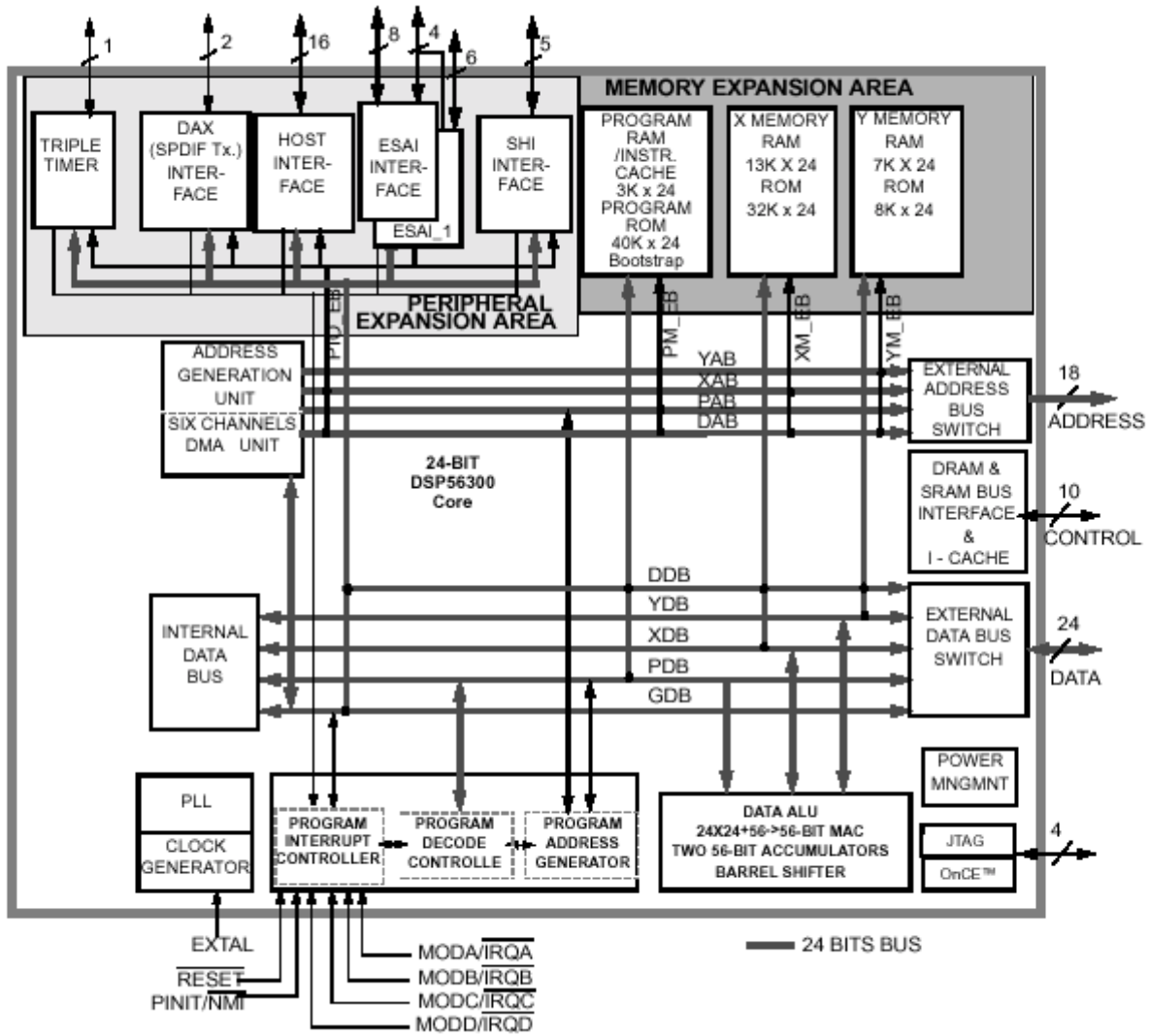


8051 Memory Architecture

- **Advantages**
  - **Simultaneous access to Program and Data store**
  - **Register banks great for avoiding context switching on interrupt and for code compression**
  - **8-bit address space extended to 256+128 = 384 registers by distinguishing between direct and indirect addressing for upper 128 bytes. Good for code compression**
  - **Bit addressable great for managing status flags**

- **Disadvantage**
  - **A little bit confusing, with potential for errors.**

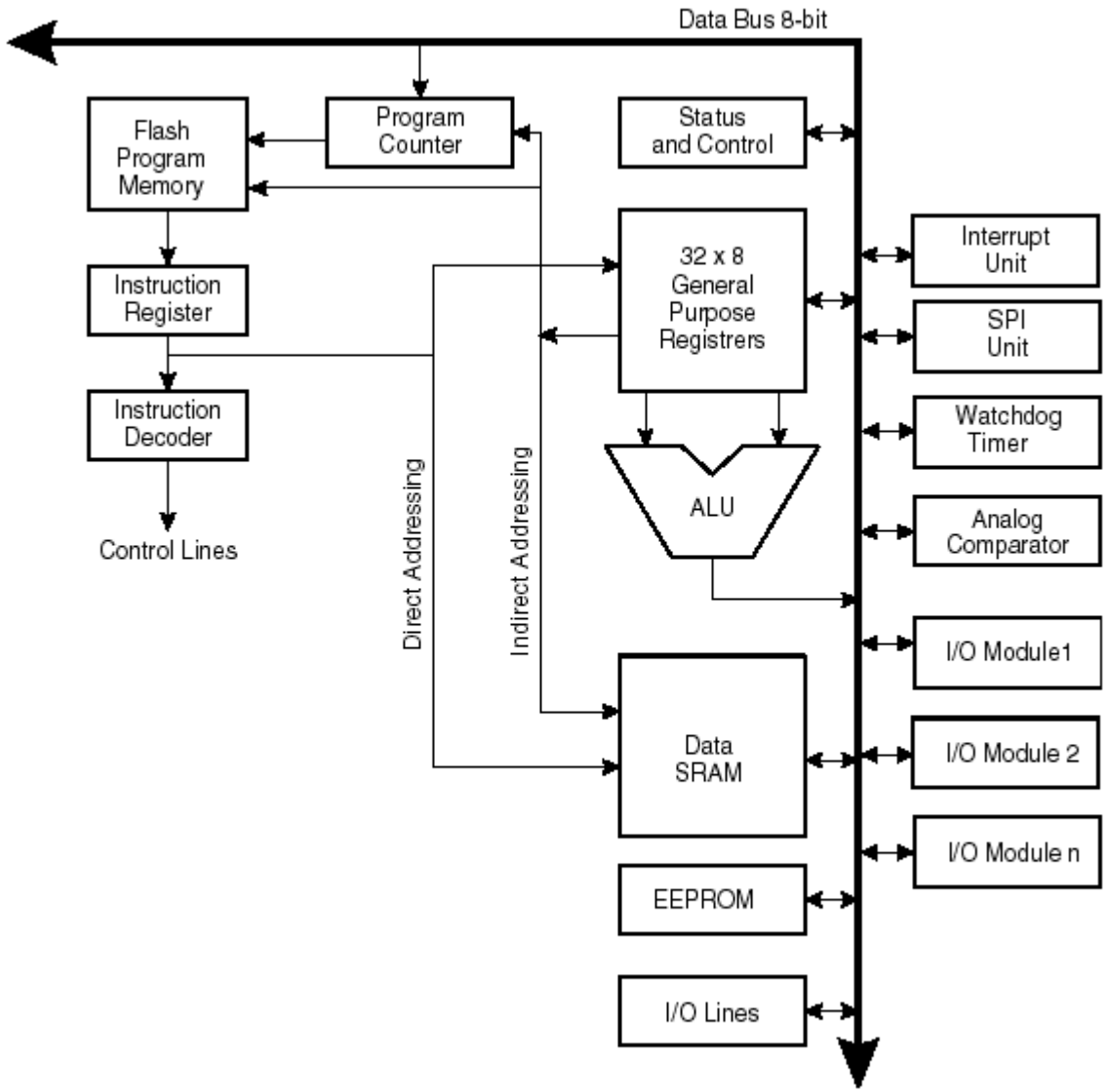
Motorola 56367 DSP



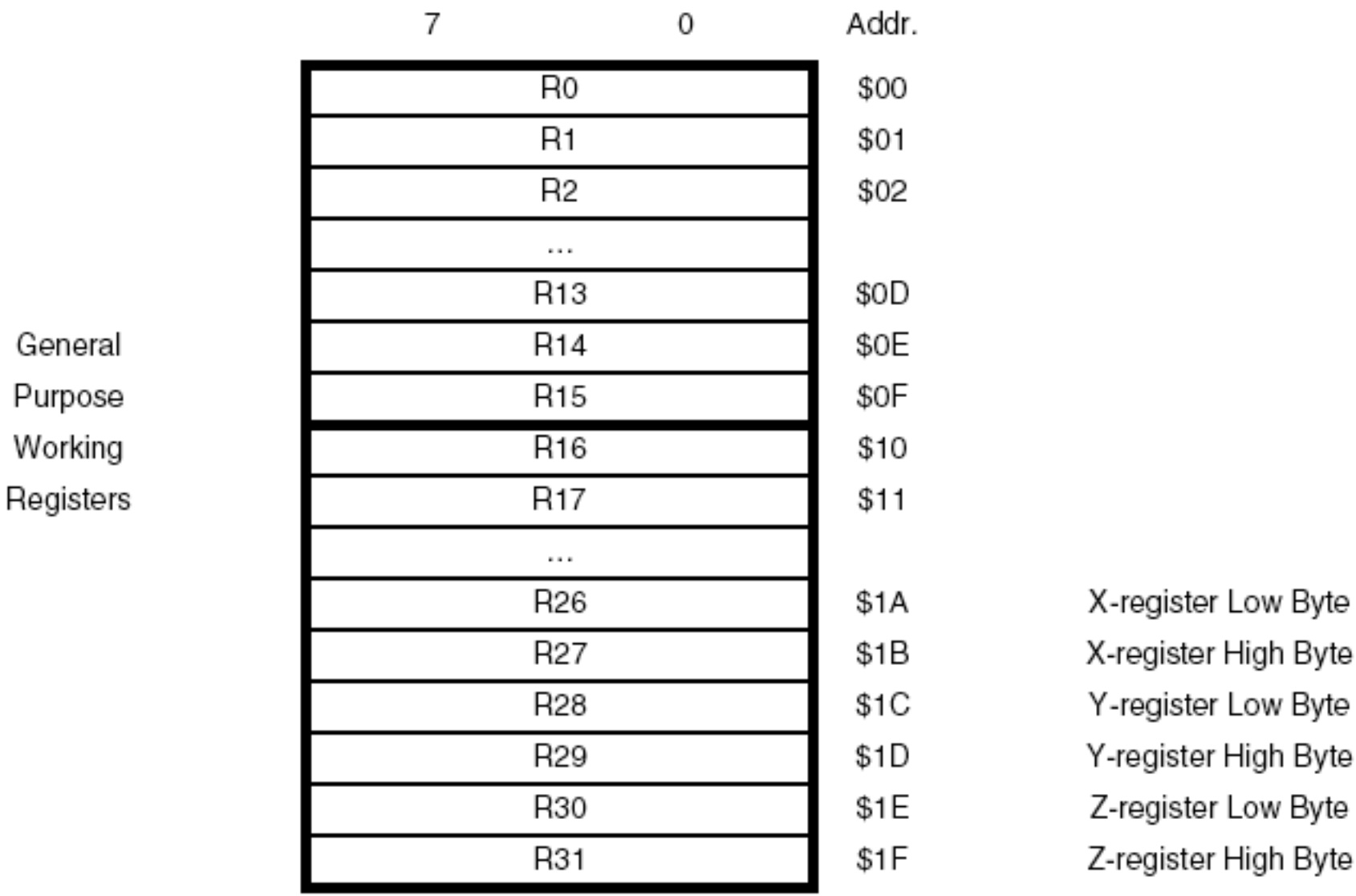
Motorola 56367 DSP

- **Advantages**
  - **Three memory spaces allow ALU op and two moves simultaneously**
    - **Macc x1,y1,a x1,x:(r3)+ y:(r6)+,y1 ; in one cycle**
    - **Really, really,really fast**
- **Disadvantages**
  - **frightening to program**
  - **Very difficult to write a good compiler**

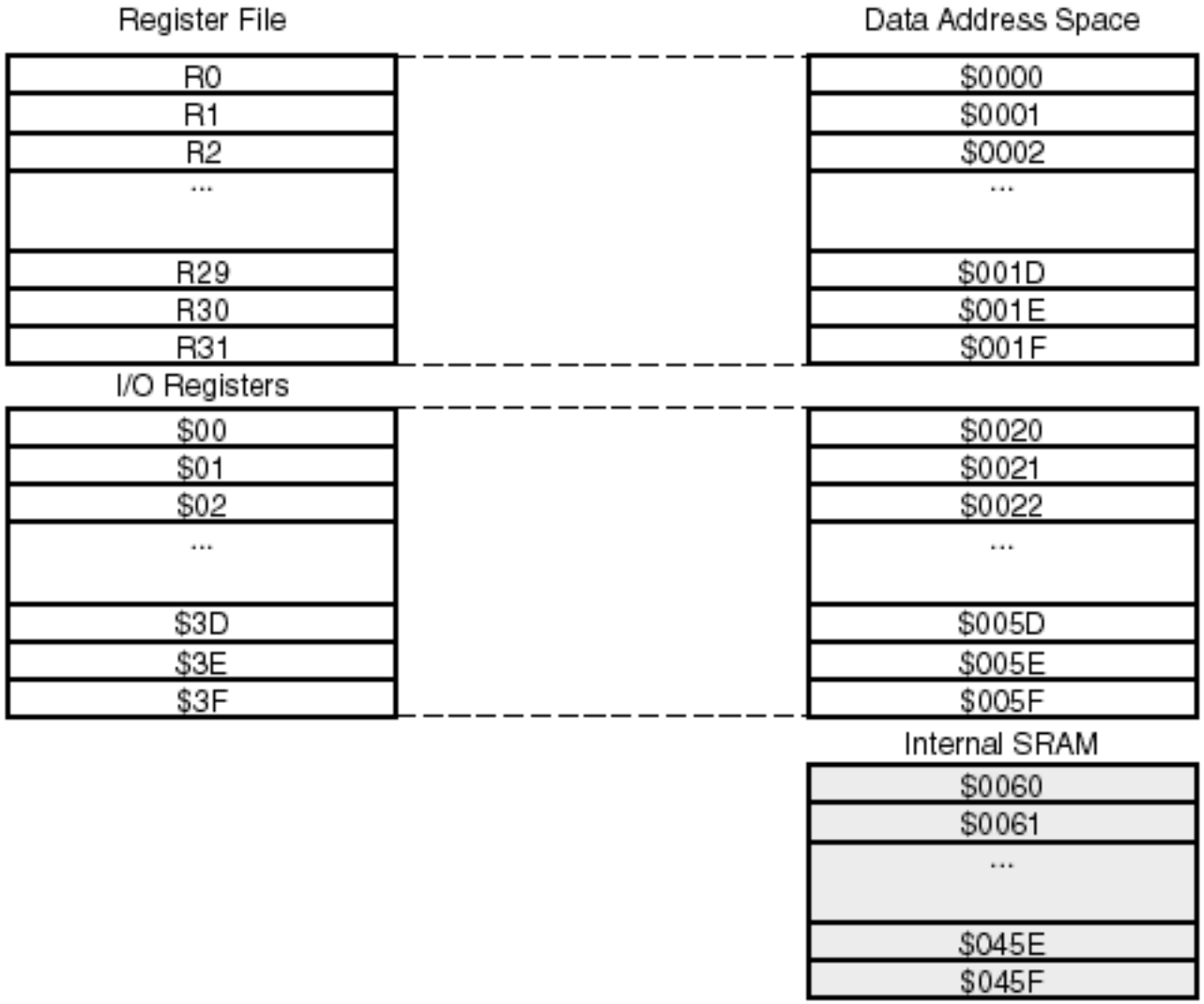
AVR CPU


Data Bus 8-bit
Flash Program Memory
Program Counter
Status and Control
Interrupt Unit
SPI Unit
Watchdog Timer
Analog Comparator
I/O Module1
I/O Module 2
I/O Module n
Instruction Register
Instruction Decoder
Control Lines
32 x 8 General Purpose Registrers
ALU
Direct Addressing
Indirect Addressing
Data SRAM
EEPROM
I/O Lines

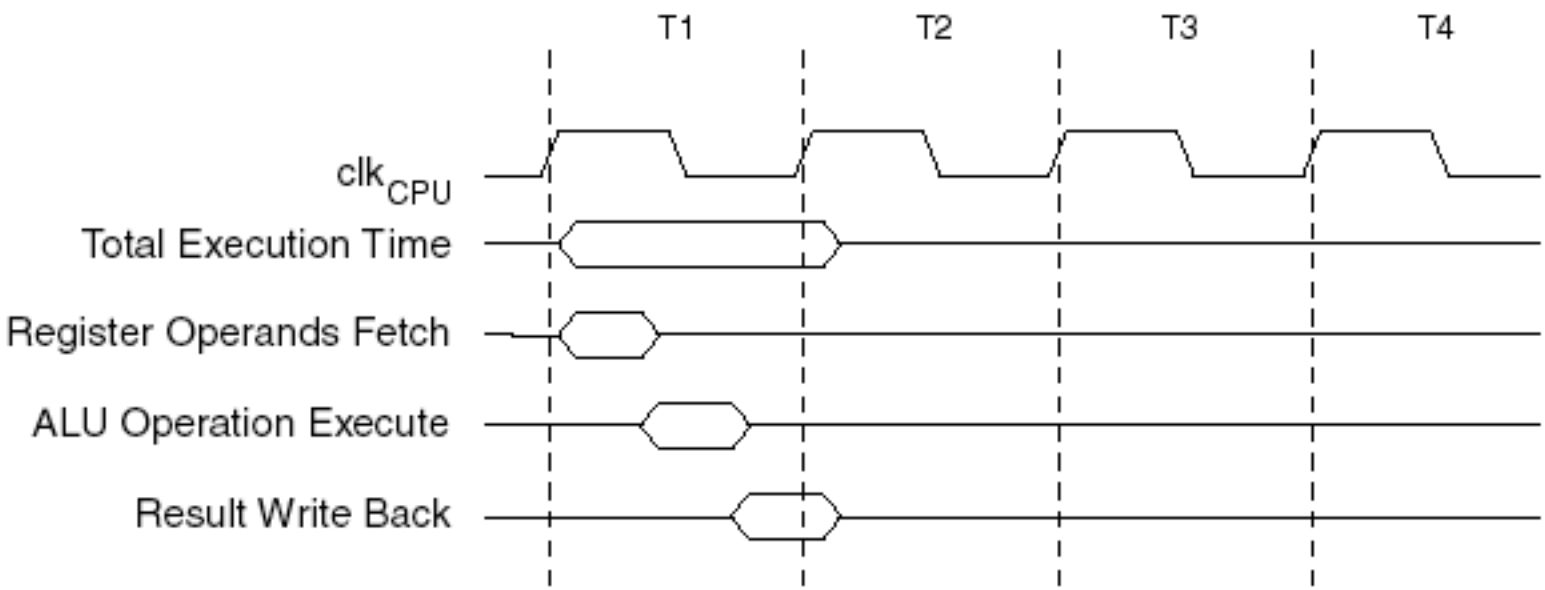
Registers


7
0
Addr.
R0
$00
R1
$01
R2
$02
...
R13
$0D
General
R14
$0E
Purpose
R15
$0F
Working
R16
$10
Registers
R17
$11
...
R26
$1A
X-register Low Byte
R27
$1B
X-register High Byte
R28
$1C
Y-register Low Byte
R29
$1D
Y-register High Byte
R30
$1E
Z-register Low Byte
R31
$1F
Z-register High Byte

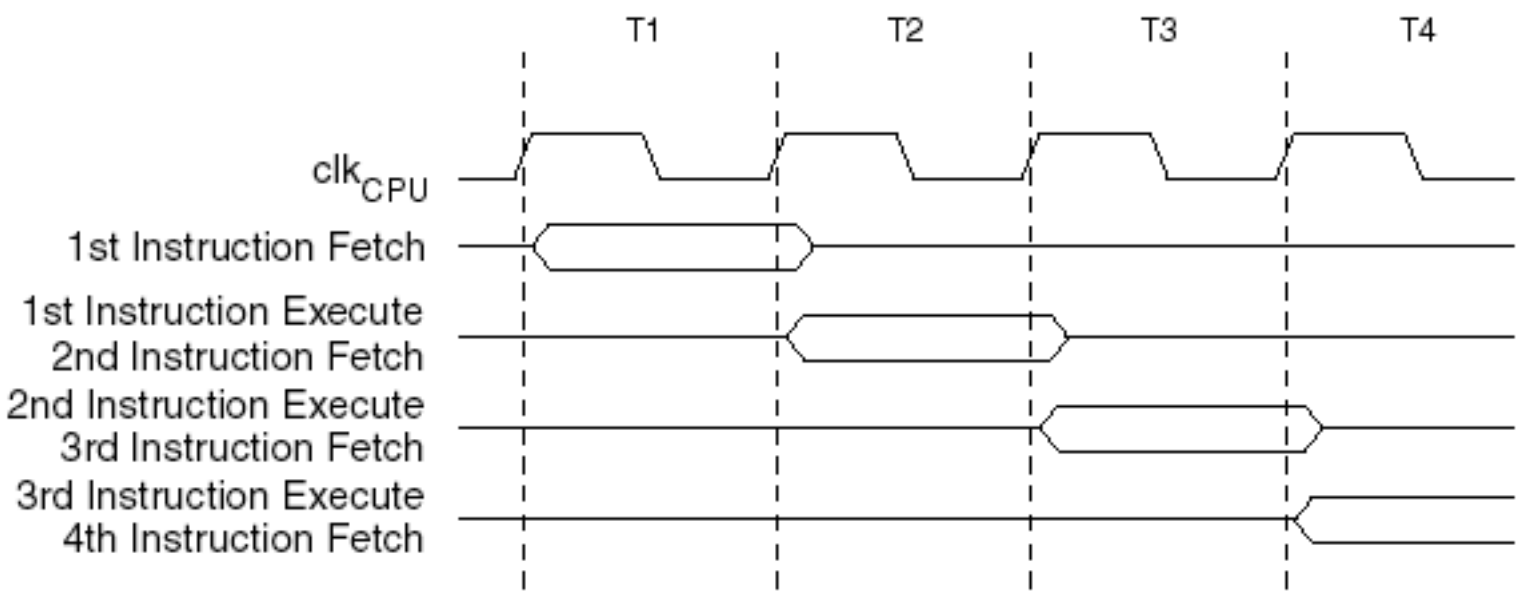
Register Mapping



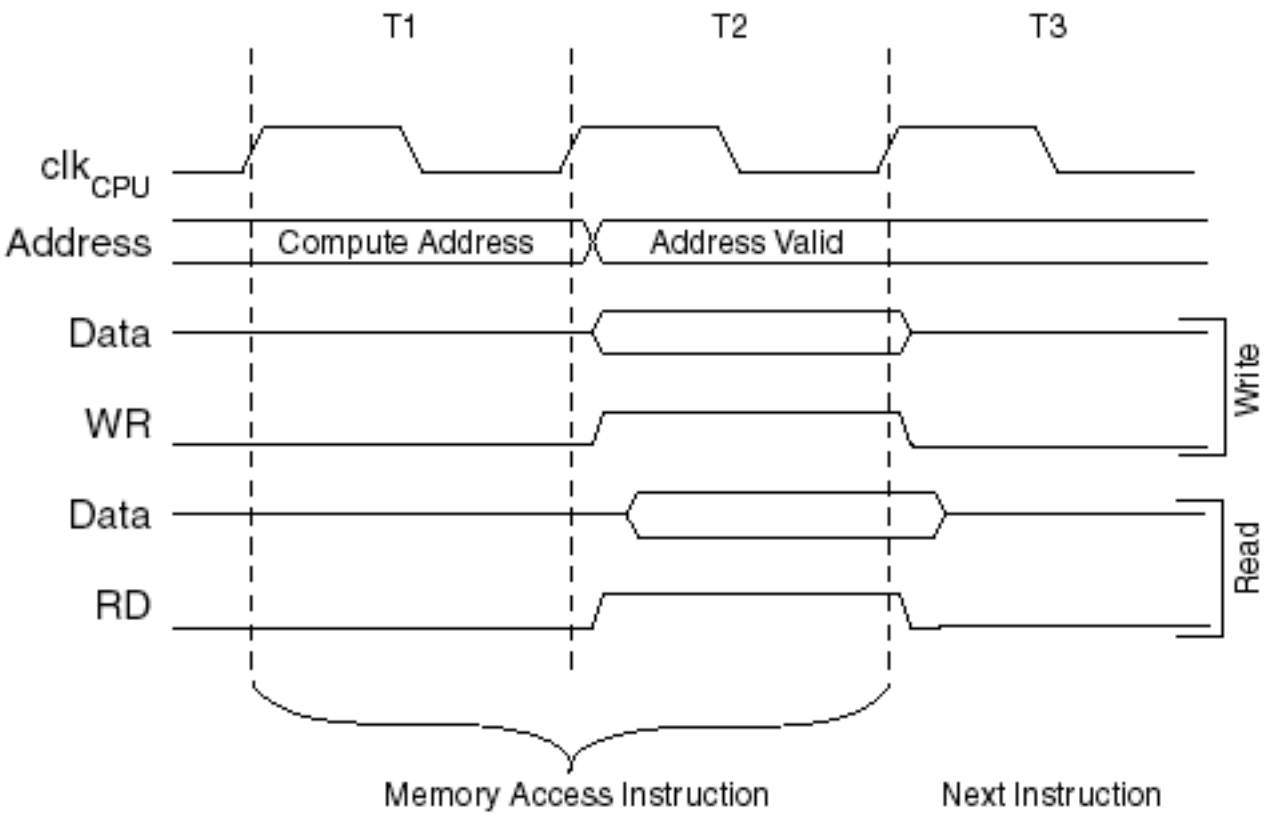
ALU Opcode Timing



Pipeline


T1
T2
T3
T4
clk CPU
1st Instruction Fetch
1st Instruction Execute
2nd Instruction Fetch
2nd Instruction Execute
3rd Instruction Fetch
3rd Instruction Execute
4th Instruction Fetch

SRAM access


T1
T2
T3
clk CPU
Address
Compute Address
Address Valid
Data
WR
Write
Data
RD
Read
Memory Access Instruction
Next Instruction

Instructions by Type

Mnemonic	Description	Mnemonic	Description	Mnemonic	Description
Flow Control		**Bit Manipulation**		**Load/Store**	
JMP ◆	Jump absolute (24-bit)	SEC/CLC	Set/clear C flag (carry)	MOV	Copy register to register
RJMP	Branch relative (12-bit)	SEH/CLH	Set/clear H flag (half carry)	LD	Load indirect through X/Y/Z
IJMP ●	Jump indirect (Z)	SEN/CLN	Set/clear N flag (negative)	LD ●	Load indirect with postincremnt
RCALL	Call subroutine	SEZ/CLZ	Set/clear Z flag (zero)	LD ●	Load indirect with predecremnt
ICALL ●	Call subroutine indirect (Z)	SEI/CLI	Set/clear I flag (interrupt)	LDD ●	Load indirect with 6-bit offset
RET/RETI	Return/from interrupt	SES/CLS	Set/clear S flag (sign)	LDI	Load 8-bit immediate
CP/CPC	Compare/with carry	SEV/CLV	Set/clear V flag (overflow)	LDS ●	Load from 16-bit address
CPI	Compare with 8-bit immediate	SET/CLT	Set/clear T bit	LPS ●	Load from program space
CPSE	Compare, skip if equal	SBR/CBR	Set/clear bit in register	ST	Store indirect through X/Y/Z
SBRS/SBRC	Skip if register bit set/clear	BSET/BCLR	Set/clear bit in status register	ST ●	Store indirect with postincremnt
SBIS/SBIC	Skip if I/O bit set/clear	SER/CLR	Set/clear entire register	ST ●	Store indirect with predecremnt
BRcc	Conditional branch	SBI/CBI	Set/clear bit in I/O space	STD ●	Store indirect with 6-bit offset
Logical		**Arithmetic**		STS ●	Store to 16-bit address
AND	Logical AND	ADD/ADC	Add/with carry	IN/OUT	Input/output to/from I/O space
ANDI	Logical AND 8-bit immediate	ADIW ●	Add 6-bit immediate	PUSH/POP	Push/pop stack element
OR	Logical OR	SUB/SUBC	Subtract/with borrow	BLD/BST	Load/store T bit
ORI	Logical OR 8-bit immediate	SBIW ●	Subtract 6-bit immediate	**Miscellaneous**	
EOR	Logical exclusive-OR	SUBI/SBCI	Subtract 8-bit imm/w borrow	NOP	No operation
LSL/LSR	Logical shift left/right by 1 bit	INC/DEC	Increment/decrement register	SLEEP	Wait for interrupt
ROL/ROR	Rotate left/right by 1 bit	MUL ◆	Multiply 8 × 8 → 16	WDR	Watchdog reset
ASR	Arithmetic shift right by 1 bit				
COM/NEG	One's/two's complement				
SWAP	Swap nibbles		Can use R16–R31 only	●	Not available on 90S1200, 1220
TST	Test for zero or minus		Can use R24–R31 only	◆	Future enhancement