

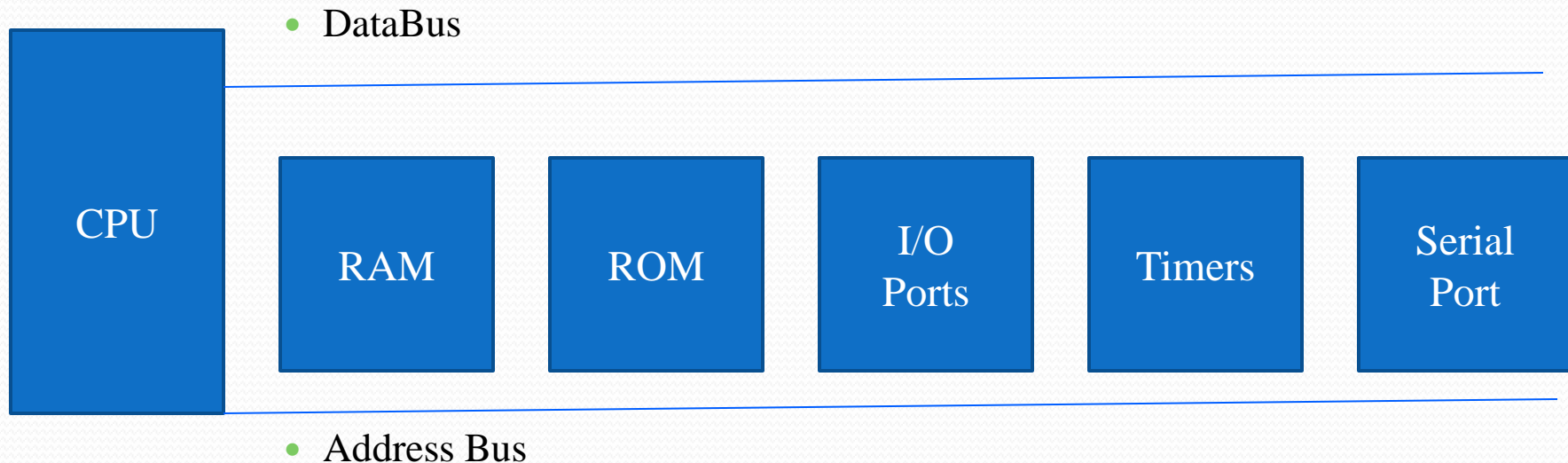
Microcontroller Introduction

Processors

- Micro-processors ?
- Microcontrollers ?

Microprocessors

- CPU for computers
- NO RAM, ROM, IO on CPU chip itself
- Example: Intel x86, Motorola 680x0



What is Microcontroller?

- Basically microcontroller is a device which integrates a number of the components of a microprocessor system onto a single microchip.
- A micro-controller combines onto the same microchip
 - The CPU core
 - Memory (both RAM and ROM)
 - Some parallel digital I/O & more

Components of a Micro-controller

- A Timer module to allow the micro-controller to perform tasks for certain time periods
- A serial I/O port to allow data to flow between the micro-controller and other devices such as a PC or another micro-controller
- An ADC to allow the micro-controller to accept analogue input data for processing

Why Micro-controller?

- Low cost, small packaging
- Low power consumption
- Programmable, re-programmable
- Lots of I/O capabilities
- Easy integration with circuits
- For applications in which cost, power and space are critical
- Single-purpose

VonNeuman Architecture

- Only one bus between CPU and memory
- RAM and program memory share the same bus and the same memory, and so must have the same width.

Harvard Architecture

- Separate program bus and databus
- Can be different widths!

CISC- Complex Instruction Set Computer

- A large number of instructions each carrying permutation of the same operation
- Instructions provide for complex operations
- Different instructions of different format
- Different instructions of different length
- Different Addressing modes
- Requires multiple cycles for execution

RISC- Reduced Instruction Set Computer

- Instructions for simple operations that can be executed in a single cycle
- Each instruction of fixed length
- Large general purpose register set
- Load-store Architecture

Manufacturers

- Intel, Freescale, Microchip (PIC), TI, Zilog
- Atmel AVR
 - Many Types, tinyAT, megaAT
 - Lighting, LCD
 - Different #s of I/O control
 - Built-in Pull-up resistors
 - Ethernet, Serial Data, USB
 - Analog I/O, Packaging, Interrupts, JTAG
 - Get the right amount of memory for the job



Atmel / AVR History

- Conceived by two students at Norwegian Institute of Technology (NTH)
 - Alf-Egil Bogen and Vegard Wollan
- Originally Known as μ RISC (Micro RISC)
- Sold to Atmel, continued working under Atmel Norway subsidiary
- AVR supposedly not acronym
 - May mean Advanced Virtual RISC

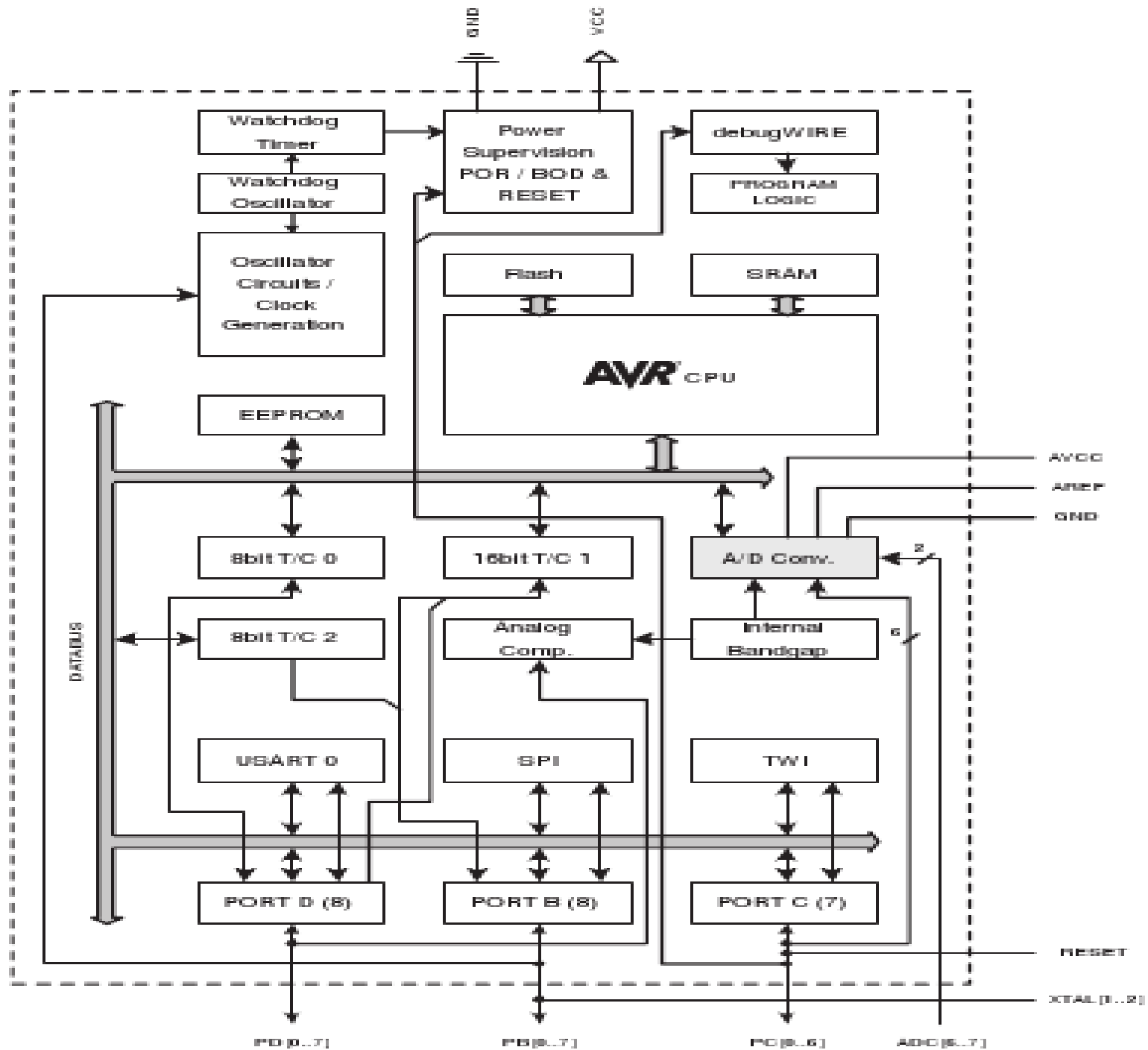
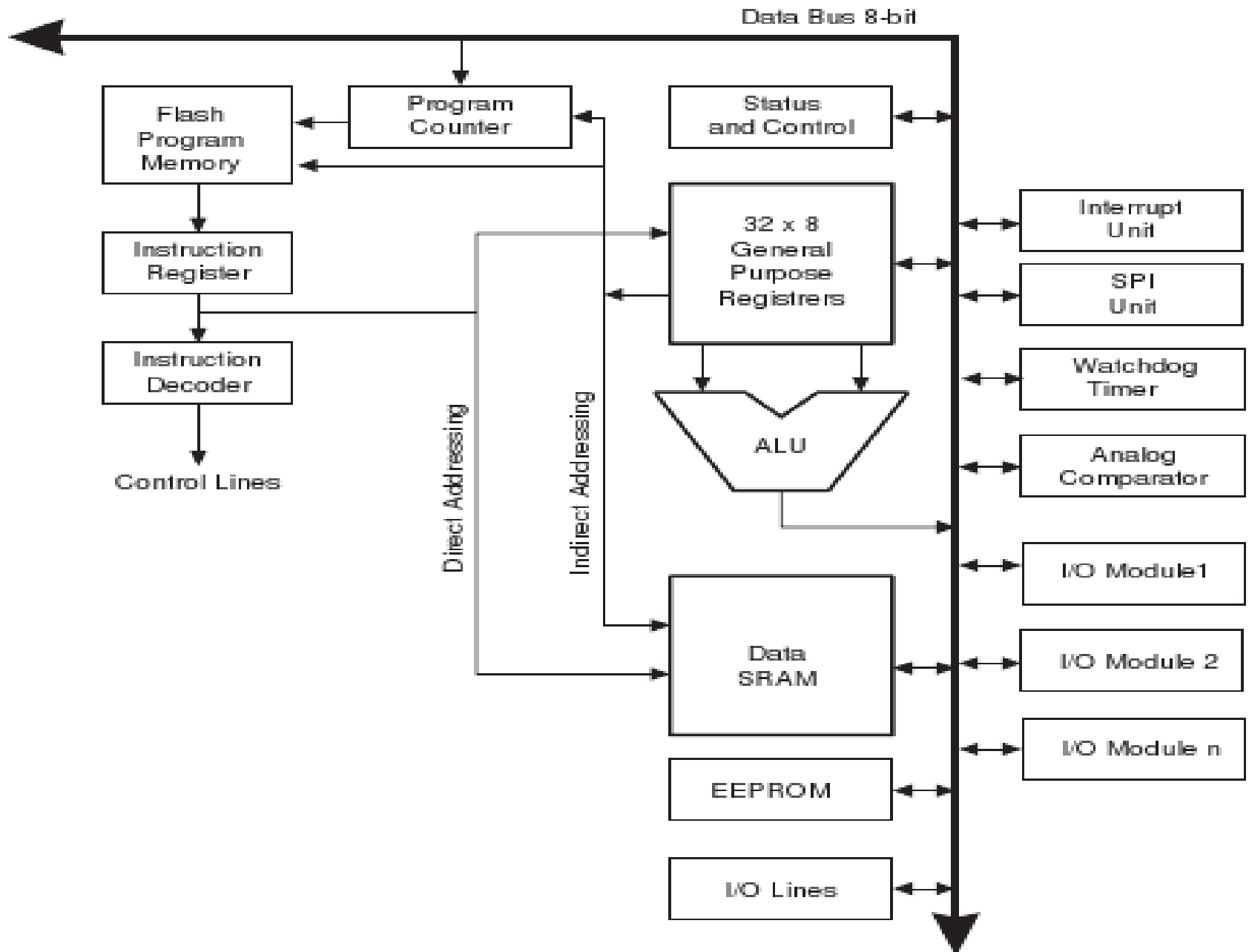


Figure 6-1. Block Diagram of the AVR Architecture



Registers

- 32 general purpose registers
- Addressable as first 32 memory addresses
- X, Y, Z registers for indirect memory
 - Dual-purpose, 16 bit
- Timers, interrupts registers
- Special function registers

Registers

Figure 6-2. AVR CPU General Purpose Working Registers

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

Memories

- 3 different memories
 - SRAM
 - Flash
 - EEPROM
- Flash is divided into two 8KB sections
 - Independently addressable

Figure 7-3. Data Memory Map

Data Memory	
32 Registers	0x0000 - 0x001F
64 I/O Registers	0x0020 - 0x005F
160 Ext I/O Reg.	0x0060 - 0x00FF
	0x0100
Internal SRAM (512/1024/1024 x 8)	0x02FF/0x04FF/0x04FF

Websites / References

- ATMEL
 - www.atmel.com
- Arduino
 - www.arduino.cc
- Adafruit Industries
 - www.adafruit.com
- FreeRTOS
 - www.freertos.org
- AVR Freaks
 - www.avrfreaks.net