AVR Introduction
The Growing AVR Family

- **TINY AVR family**
  - 8 - 32 pin general purpose microcontrollers
  - 16 family members

- **MEGA AVR family**
  - 32 - 100 pin general purpose microcontrollers
  - 23 family members

- **ASSP AVRs**
  - USB, CAN and LCD
  - Motor Control and Lighting
  - Automotive
  - Battery Management
  - 8 family members
Scalable

- Devices range from 1 to 256KB
- Pin count range from 8 to 100
- Full code compatibility
- Pin/feature compatible families
- One set of development tools

= Roadmap for the future
AVR Architecture

- RISC architecture with CISC instruction set
  - Powerful instruction set for C and Assembly
- Scalable
  - Same powerful AVR core in all devices
- Single cycle execution
  - One instruction per external clock
  - Low power consumption
- 32 Working Registers
  - All Directly connected to ALU!
- Very efficient core
  - 20 MIPS @ 20MHz
- High System Level Integration
  - Lowest total system cost
Register operations take ONE clock pulse on the EXTERNAL clock input.

20MIPS @ 20MHz
High Code Density

- Architecture designed for C
- 32 general registers
- C-like addressing modes
- 16- and 32-bit arithmetic support
- Linear address maps
AVR Influenced by IAR

- Architecture and Instruction Set co-designed with IAR systems through several iterations:
  - Compiler development project initiated before architecture and instruction set frozen
  - Compiler experts’ advice implemented in hardware
  - Potential HLL bottlenecks identified and removed

Auto Increment/Decrement Example:

C Source:

```c
unsigned char *var1, *var2;
*var1++ = *--var2;
```

Generated assembly code:

```
LD R16,-X
ST Z+,R16
```
/* Return the maximum value of a table of 16 integers */

int max(int *array)
{
    char a;
    int maximum=-32768;

    for (a=0;a<16;a++)
        if (array[a]>maximum)
            maximum=array[a];
    return (maximum);
}
MSP430 and AVR are running a close race
  - But max speed on MSP430 is only 8MHz

The C51 would have to run at 296 MHz to match the 16 MHz AVR

PIC18 seems fast but require 3 times as much code space.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ATmega16</td>
<td>16</td>
<td>32</td>
<td>227</td>
<td>14.2</td>
</tr>
<tr>
<td>MSP430</td>
<td>8</td>
<td>34</td>
<td>246</td>
<td>30.8</td>
</tr>
<tr>
<td>T89C51RD2</td>
<td>20</td>
<td>57</td>
<td>4200</td>
<td>210.0</td>
</tr>
<tr>
<td>PIC18F452</td>
<td>40</td>
<td>92</td>
<td>716</td>
<td>17.9</td>
</tr>
<tr>
<td>PIC16C74</td>
<td>20</td>
<td>87</td>
<td>2492</td>
<td>124.6</td>
</tr>
<tr>
<td>68HC11</td>
<td>12</td>
<td>59</td>
<td>1238</td>
<td>103.2</td>
</tr>
</tbody>
</table>
Benchmark: Real-life Applications

• Complete navigation application
• C bitfields
• Car Radio control
• DES encryption / decryption
• Three different modules from analog telephones
• Reed-Solomon (error correction) encoder/decoder
• Pager protocol
• Refrigerator control
• Battery charger
• Embedded web server
• Label/recite printer
Benchmark – Code size
High Integration

AVR core

Digital

NV-memory

4-Wire In/Out

3-Wire In/Out

JTAG Interface

Flash

Serial Peripheral Interface

EEPROM

Analog

Serial Interfaces

Watchdog Timer

I/O Ports

Interrupts

Timer/Counters

Analog Comparator

A/D Converter

LCD Interface

USART

SPI

TWI
AVR – Single-Chip Solution

Many uC only give you this
AVR – Single-Chip Solution

AVR Integrates Much More!

- TWI
- USART
- SPI
- Hardware Multiplier
- Flash
- EEPROM
- Analog Comparator
- CPU CORE
- SRAM
- A/D Converter
- Register File
- I/O pins
- Brown Out Detector
- Analog Reference
- Pull-Ups On Demand
- Reset Circuitry
- High Current Outputs
- Programmable Watchdog
- Calibrated Oscillator
- On-Chip Debug
- In System Programming
- JTAG
- Boundary Scan
- LCD Interface
- LCD DISPLAY
- Temperature Sensor
- AV Integrated

www.atmel.com
Highest System Level Integration
In-System Development

- In-System Programming
- In-System Debugging
- In-System Verification
AVR Self-Programming

• Redefining ISP → Self-Programming
  • The AVR reprograms itself
  • Any existing communication interface
  • Any voltage
  • Any frequency

• Critical functions still operating
  • Run code during programming (Read-While-Write)

• Software controlled programming
  • Firmware updates
  • Parameter updates
The Flash program memory is divided into two sections:
- Application Section
- Boot Loader Section

The two sections enable the AVR to handle two independent applications:
- The Application section contains the main application
- The Boot Loader section contains a Flash programming application

Note that small AVRs do not divide the Flash:
- The whole Flash can be considered as a Boot Loader
- Only on devices with 4K Flash or less
Self-Programming Flexibility

- AVR Self-Programming is controlled by SW
  - SPM instruction controls self-programming
  - SPM is an AVR core feature
  - Not a hard-coded firmware, but a part of the customer application

- The AVR updates its own Flash while running
  - Similar to AVR EEPROM access
  - Critical functions in the customers application can be maintained

- The upgrade data can be received from any interface
  - No restricted communication protocol or interface
  - No external hardware

- No restrictions to Vcc or Clock frequency
All AVRs have In System Debugging

- JTAG interface
  - On High pin-count devices
  - Uses 4 general I/O pins

- debugWIRE interface
  - On Low pin-count devices
  - Uses only Reset pin

Compared to JTAGs four pins, debugWIRE uses only one; Reset. This is a big advantage on low pin count devices
Development Tools

• AVR Studio - front end for all AVR tools
  • Free

• Starter kits and evaluation boards
  • From $19

• On-Chip Debuggers and Emulators
  • From $299
Support

- Fully updated product web
- Highly skilled Field Application Engineers
- Support mail handled by AVR experts
- Reference designs
- Application notes
- AVRfreaks community website

= Ensures no slip in schedule
Summary

- High performance
- Low power consumption
- High code density
- Advanced memory technology
- High integration

= Leading 8-bit microcontroller