Integrated Circuits in 1958
The first single-chip microprocessor, circa 1971

(Intel 4004: 740kHz, 4-bit, 4KB cache, 2,300 transistors)
microprocessors, circa 2009

(Intel Tukwila: quad-core, 2GHz, 64-bit, 30MB cache, 2.0B transistors)
28-pin DIP
8-bit Atmel ATmega32U4 processor
25 I/O pins
12 channel 10-bit ADC
32k Flash, 1k EEPROM, 2.5k SRAM
4 independent timers with PWM
USART, I2C, SPI, JTAG, USB
The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega16U4/ATmega32U4 provides the following features: 16/32K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512Bytes/1K bytes EEPROM, 1.25/2.5K bytes SRAM, 26 general purpose I/O lines (CMOS outputs and LVTTL inputs), 32 general purpose working registers, four flexible Timer/Counters with compare modes and PWM, one more high-speed Timer/Counter with compare modes and PLL adjustable source, one USART (including CTS/RTS flow control signals), a byte oriented 2-wire Serial Interface, a 12-bit Analog to Digital Converter, an 8-bit Data Path, an 8-bit Wide External Bus, an Arithmetic Logic Unit (ALU), a Program Counter operating in 8-bit or 16-bit mode, a Stack Pointer, a Status Register, a RESET mechanism, a Clock Oscillator System, a PLL Clock System, a Watchdog Timer, a Power-On Reset circuit, an On-Board Oscillator Calibration Circuit, a Boundary-Scan Circuit, a JTAG boundary-scan and an On-Chip Debug Circuit.
Microcontroller Architectures

A Von Neumann architecture computer uses a shared pathway for instructions and data to/from the CPU

A Harvard architecture computer uses separate pathways for instructions and data
Memory Map

32k Flash
“Program Memory”

0x0000

instruction
(16 or 32 bit)

CPU

data
(8 bit)

address
(16 bit)

address
(16 bit)

2.5k SRAM
“Data Memory”

0x0000

0x0AFF

0x7FFF

© 2009, Jonathan Fiene, University of Pennsylvania
# 8-bit Data Memory

<table>
<thead>
<tr>
<th>Address</th>
<th>Memory Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000 - 0x0020</td>
<td>32B</td>
</tr>
<tr>
<td>0x0060 - 0x0100</td>
<td>60B</td>
</tr>
<tr>
<td>0x0100 - 0x0AFF</td>
<td>160B</td>
</tr>
<tr>
<td>0x0AFF</td>
<td>2.2KB</td>
</tr>
</tbody>
</table>

- **working reg**
- **I/O**
- **ext. I/O**
- **SRAM**
Random Access Memory

<table>
<thead>
<tr>
<th>working reg</th>
<th>I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ext. I/O</td>
</tr>
<tr>
<td></td>
<td>SRAM</td>
</tr>
</tbody>
</table>

program variables translate to specific addresses in RAM

address: 0x02A5

value: 0x2F

0x0100

0x0AFF
Peripherals

working reg

I/O

ext. I/O

SRAM

GPIO

Clock

Serial Comm

Timer

A2D

0x0020

0x00FF

Pre-defined names translate to specific addresses in the registers