Appendix E - Embedded Processors
<table>
<thead>
<tr>
<th>Feature</th>
<th>Desktop</th>
<th>Server</th>
<th>Embedded</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price of system</td>
<td>$1000–$10,000</td>
<td>$10,000–$10,000,000</td>
<td>$10–$100,000 (including network routers at the high end)</td>
</tr>
<tr>
<td>Price of microprocessor module</td>
<td>$100–$1000</td>
<td>$200–$2000 (per processor)</td>
<td>$0.20–$200 (per processor)</td>
</tr>
<tr>
<td>Microprocessors sold per year (estimates for 2000)</td>
<td>150,000,000</td>
<td>4,000,000</td>
<td>300,000,000 (32-bit and 64-bit processors only)</td>
</tr>
<tr>
<td>Critical system design issues</td>
<td>Price-performance, graphics performance</td>
<td>Throughput, availability, scalability</td>
<td>Price, power consumption, application-specific performance</td>
</tr>
</tbody>
</table>

Figure E.1 A summary of the three computing classes and their system characteristics. Note the wide range in
Characteristics of Embedded Processors

Deterministic Instruction Execution

- Sacrifice performance for predictable execution behavior
- Each instruction has specific timing characteristics
- Simple memory structures (predictable cache behavior)
- Simplified instruction fetch, simple pipelines
- Tomasulo execution?
Characteristics of Embedded Processors

Calculation behavior

• Saturation arithmetic - rather than value wraparound, “overflow” values clamped at maximum.

• Fixed point arithmetic - integer arithmetic, radix point is assumed.
  • Compared with floating point, sacrifices dynamic range for execution speed

• Special instruction types
  • Multiply-Accumulate (MAC) instruction
  • Built-in sine/cosine function approximations
  • SIMD Instructions
Characteristics of Embedded Processors

Plentiful I/O

• I/O Pins are a precious resource!

• Pins have multiple functions
  • Input
  • Output
  • Current sink/source
  • Standard interfaces (I2C, USB, PWM, etc)

• Memory Address/Data

• Interrupts
Embedded Categories

- Microcontrollers
  - 8- 32 bit data path
  - No/limited operating system
  - Power efficient
  - Flexible I/O capabilities
  - Both legacy and modern architectures
  - Built-in flash memory for instructions, RAM for data
  - Examples: 8051, 8080/Z80, AVR (Arduino), PIC (Microchip)
Embedded Categories (Con’t)

• Microprocessors
  • 8-64 bit data path
  • Real time operating system (Linux, Free RTOS, Lynx-OS)
  • Power-efficient
  • Flexible I/O
  • Built-in Flash ROM, RAM
• Examples: Raspberry-PI, 80x86, ARM
Embedded Categories (Con’t)

• Digital Signal Processors

  • 16-64 bit data path

  • Common/RTOS/specialized Operating system (Linux, DSP/BIOS)

  • Regular and specialized instructions such as MAC (Multiply/accumulate)
    • FFT (Fast Fourier transform)

\[
X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad \text{where} \quad W_N^{kn} = e^{j\frac{2\pi kn}{N}} = \cos\left(\frac{2\pi kn}{N}\right) + j\sin\left(\frac{2\pi kn}{N}\right)
\]

• Power efficient

• Flexible I/O

• Examples: TI320C55, TI 320C6x, DSP56301
Embedded Categories (Con’t)

• High performance processors
  • 24-64 bit data path
  • Specialized versions of common processors
  • Variety of Operating Systems (Linux, Windows, RTOS’s)
  • Deterministic execution
  • Examples: Pentium, ARM
AVR Processor
Microcontroller

- 8-bit data path, 16 bit instructions
- 2-stage pipeline - fetch, execute
- 16 MHz clock speed
- Built-in Flash ROM for instructions, RAM for memory
- Family includes 8-pin package, <$1.00 cost
- Basis for Arduino UNO, ~$25, 1K RAM, 32K ROM
Raspberry PI
Microprocessor

- ARM-based single board computer
- 80 MHz clock speed
- Linux-capable
- Memory management
- Some models are multi-core
- Some models <$25.00
<table>
<thead>
<tr>
<th>Generation</th>
<th>Year</th>
<th>Example DSP</th>
<th>Data width</th>
<th>Accumulator width</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1982</td>
<td>TI TMS32010</td>
<td>16 bits</td>
<td>32 bits</td>
</tr>
<tr>
<td>2</td>
<td>1987</td>
<td>Motorola DSP56001</td>
<td>24 bits</td>
<td>56 bits</td>
</tr>
<tr>
<td>3</td>
<td>1995</td>
<td>Motorola DSP56301</td>
<td>24 bits</td>
<td>56 bits</td>
</tr>
<tr>
<td>4</td>
<td>1998</td>
<td>TI TMS320C6201</td>
<td>16 bits</td>
<td>40 bits</td>
</tr>
</tbody>
</table>

Figure E.2 Four generations of DSPs, their data width, and the width of the registers that reduces round-off error.
Digital Signal Processor - TI320C55

Seven Stage Pipeline

- *Fetch stage* reads program data from memory into the instruction buffer queue.
- *Decode stage* decodes instructions and dispatches tasks to the other primary functional units.
- *Address stage* computes addresses for data accesses and branch addresses for program discontinuities.
- *Access 1/Access 2 stages* send data read addresses to memory.
- *Read stage* transfers operand data on the B bus, C bus, and D bus.
- *Execute stage* executes operation in the A unit and D unit and performs writes on the E bus and F bus.
Figure E.4 Architecture of the TMS320C55 DSP. The C55 is a seven-stage pipelined processor with some unique instruction execution facilities. (Courtesy Texas Instruments.)
Figure E.5 Architecture of the TMS320C64x family of DSPs. The C6x is an eight-issue traditional VLIW processor. (Courtesy Texas Instruments.)
<table>
<thead>
<tr>
<th>Instruction</th>
<th>31</th>
<th>0 31</th>
<th>0 31</th>
<th>0 31</th>
<th>0 31</th>
<th>0 31</th>
<th>0 31</th>
<th>0 31</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
</tr>
<tr>
<td>B</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
</tr>
<tr>
<td>C</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
</tr>
<tr>
<td>D</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
</tr>
<tr>
<td>E</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
</tr>
<tr>
<td>F</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
</tr>
<tr>
<td>G</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
</tr>
<tr>
<td>H</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>p</td>
</tr>
</tbody>
</table>

**Figure E.6 Instruction packet of the TMS320C6x family of DSPs.** The \( p \) bits determine whether an instruction begins a new VLIW word or not. If the \( p \) bit of instruction \( i \) is 1, then instruction \( i + 1 \) is to be executed in parallel with (in the same cycle as) instruction \( i \). If the \( p \) bit of instruction \( i \) is 0, then instruction \( i + 1 \) is executed in the cycle after instruction \( i \). (Courtesy Texas Instruments.)