Contents

- Raspberry Pi Foundation
- Raspberry Pi overview & specs
- ARM11 overview
- ARM11 cache, pipeline, branch prediction
- ARM11 vs. ARM9
- ARM11 future
- Programming the Raspberry Pi
- Raspberry Pi uses and project ideas
- Raspberry Pi vs. Arduino
Raspberry Pi Foundation

- UK registered educational charity
- Promote the study of computer science
- Cheap, low-power, programmable computers
- Education, developing nations
- One Laptop per Child, Arduino
Raspberry Pi

- ultra-low-cost ($25-$35)
- credit-card sized Linux computer
- 10/100 Ethernet, HDMI, SD-card, USB
- GPIO to control external devices
- UART, I²C interface
Specifications

- Broadcom BCM2835 System on a chip (SoC) [1]
  - ARM1176JZF-S processor
    - 700 MHz with overclock capability to 1 GHz
  - VideoCore IV GPU [2]
    - Supports 1080p H.264 decoding and encoding
  - 256 MB (model A) or 512 MB (model B) SDRAM [3]
- 26 available low-level peripherals
  - GPIO, UART, I2C, I2S, +3.3V, +5V, GND
- Price: $35
ARM11- Overview

- Based on the ARMv6 instruction set architecture. [4]
  - 32-bit ARM ISA
  - 16-bit Thumb ISA
    - Used in embedded systems with low amounts of memory, not used in the RPi.
- 16 General-purpose registers Ro-R15, 15 general purpose “mode-specific” registers, 7 status registers [5]
- Designed to run various operating systems (Unix/Linux, Android) on many different devices (iPhone (Gen 3), Nintendo 3DS, etc.)
ARM11 Cache Organization

- Four-way set associative cache of configurable size. [4]
  - Virtually indexed and physically tagged
  - Sizes in the range of 4 to 64KB
- Processor level-one memory has separate instruction and data caches
  - Data cache misses are non-blocking [5]
- L1 cache’s write buffer handles all RAW hazards
- L2 cache is off chip
Eight Stage Pipeline

- **Fe1**: Address is issued to memory, data returns from memory. [4]
- **Fe2**: Branch prediction
- **De**: Instruction Decode
- **Iss**: Register read and instruction issue
- **Sh**: Shifter stage
- **ALU**: Main integer operation calculation
- **Sat**: Enable saturation of integer results
- **Wbex**: Write back
ARM1176JZF-S Pipeline Stages
Typical Multiply Pipeline
Branch Prediction \[6\]

- **Static Predictor**- First time run
  - Forward conditional branches are assumed to be taken
  - Backward conditionals are assumed to be not taken

- **Dynamic Predictor**-
  - Branch prediction cache as 128 entries based on the branch instruction address.
  - States: Strongly not taken, weakly not taken, strongly taken, weakly taken.

- A **return stack** predicts unconditional subroutine return instructions.
ARM11 vs. ARM9

- Cache is physically addressed in ARM11[7]
  - ARM9 had cache aliasing and context switching issues.
- Redesigned pipeline
  - 8 stages for ARM11 vs. 5 for ARM9
  - ARM11 has out-of-order completion
  - No dynamic branch prediction in ARM9 (always assumed not taken)
  - ARM11 has Load/Store and ALU parallelism
- ARM11 64-bit data paths vs. 32-bit ARM9
ARM11 Future

- Very little.
- The reason that the Raspberry Pi is extremely cheap is because the ARM11 is outdated.
- It’s successor, the ARM Cortex A8 is a dual-issue superscalar design, achieving double the IPC. [8]
  - The A8 is seen in the iPhone 4
  - The A8 is been also seceded by the Cortex A15 and can be seen in the Nvidia Tegra 4 [9]
Programming Raspberry Pi

- Raspbian, Arch Linux ARM, Debian, Android
- Chrome OS, RISC OS, Fedora
- Python, C, Perl, Java (must compile for ARMv6)
- Education Manual
- Scratch - visual programming environment
- Raspberry Pi App store [10]
Raspberry Pi uses

- University of Southampton supercomputer [11]
- Programming introduction for children
- Senior Design projects
### Raspberry Pi vs. Arduino

<table>
<thead>
<tr>
<th>Raspberry Pi</th>
<th>Arduino</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microprocessor</td>
<td>Microcontroller</td>
</tr>
<tr>
<td>Computer science</td>
<td>Electronics</td>
</tr>
<tr>
<td>multimedia</td>
<td>Signal processing</td>
</tr>
<tr>
<td>Full Linux environment</td>
<td>Real time applications</td>
</tr>
<tr>
<td>Python, C, Perl, Java</td>
<td>C or C++</td>
</tr>
</tbody>
</table>
References


