The First Licensable, Clockless 32-bit Processor Core

Arjan Bink, ARM996HS Technical Lead, Handshake Solutions
Richard York, CPU Product Manager, ARM
ARM - Handshake Solutions Partnership

- Announced in October 2004
- Jointly develop ARM core implementations
  - Enables new classes of applications where needs for ultra-low power, low electromagnetic emissions, and robustness converge
- ARM996HS announced in February 2006
  - Jointly market and promote the ARM996HS
  - Licensing of the ARM996HS is done by ARM Ltd
- Potential application domains
  - Automotive
  - Low-cost consumer electronics
  - Wireless
  - Medical implants
  - Smartcards
  - Sensor networks
Handshake Technology

- **Design flow for clockless digital ICs**
  - C and behavioural Verilog-like design-entry language called HASTE
  - Based on a library of handshake components
  - Uses local handshakes instead of a global clock
  - Supports integration with synchronous blocks and systems

- **Industry-proven tools and flow**
  - 25 chip designs on the market
  - Over 100 million ICs already sold

- **The ARM licensee receives:**
  - Targeted Verilog netlist
  - Back-end scripts

- **The flow to produce the netlist is completely invisible to the ARM licensee**
Handshake Technology Inside

- Modules communicate by means of handshakes
- Handshakes consist of alternating request and acknowledge signals
- Request and acknowledge may contain (encode) data
- Four-phase handshake signaling
- Single-rail data encoding (bundled data)
- Thoroughly tested and completely hidden from the user
Handshake Technology Netlists

Control
Asynchronous with Muller-C elements

Datapath

Data in

Latches
logic block 1
Flip-flops
logic block 2
Latches

Data out

Reset
Handshake signals
delay 1
delay 2
ARM996HS Overview

- 32-bit RISC CPU core
- ARMv5TE architecture
- Five-stage integer pipeline
- Fast 32-bit MAC
- 16-bit Thumb® and 32-bit ARM instruction sets
- Harvard bus architecture
- Dual AMBA™ 3 AHB-Lite™ interfaces
  - Instruction interface
  - Data interface
- Memory-protection unit (MPU)
- Nonmaskable interrupts (NMI)
- Hardware divide
**ARM996HS Pipeline**

- Typical ARM9E five-stage pipeline implementing the ARMv5TE ISA
- Pipeline control ensures *distributed* stage *activation*, *parallel execution*
- Stages clock only the required data elements
- Pipeline handshakes with system controller for instruction fetches, loads, and stores
Dual AMBA™ 3 AHB-Lite™

- **Dual** AMBA 3 AHB-Lite interfaces
  - Instruction interface
  - Data interface

- Improved system performance
  - Each stream has a dedicated bus

- Greater flexibility of system-level architecture
  - System architect decides where and how to unify

- Fully synchronous
  - Allows easy integration into a synchronous ASIC
Enhanced Memory-Protection Unit

- **32-byte granularity** for fine-grain protection
  - Important for stack checking

- **Unified 12-region MPU**
  - Eight words or greater

- **Separates:**
  - User from system
  - Task from task
  - Data from data
  - Stack from stack

- **Allows overlapping regions**
  - For shared access

---

Example of protected memory map
Hardware Divide

- Supports signed and unsigned 32-bit division
  - `UDIV` and `SDIV` instructions
- Implemented as an internal coprocessor 7 (c7)
  - Three user-accessible registers
  - Based on a 16-iteration SRT division algorithm
- Operates in parallel with the pipeline
  - Code can take advantage of that fact
- 13 equivalent cycles vs. 36 for ARM968E-S
- Straightforward tools and library support
  - Simple replacement for existing real-time divide library code
Nonmaskable interrupts

- A **top-priority interrupt** that cannot be masked by software
- Mapped onto the nFIQ interrupt signal
- Ideal for embedded-control applications where **high reliability** or **high availability** are paramount
- Typical uses: indicating parity failure in memory or critically low energy level in batteries
Simple read-access example

Handshake allows core to automatically adapt to native speed of RAM

ARM996HS uses the RAM Clock (CL) and Ready (RY) pins to handshake with the RAM

- Simple TCM configuration example
- Integrates with off-the-shelf synchronous RAM
- No glue logic needed
Handshake circuits automatically adapt to changes in environmental conditions:
- Temperature, supply voltage, supply current

- Very robust
  - Continues operating correctly over ranges in which a synchronous core could break down

- No way to slow down the circuit to mimic worst-case environment conditions
  - Circuit performance depends on the operating conditions
Solution: HT-Metrics Peripheral

- HT-Metrics acts as a brake for the ARM996HS via the Tempo interface
  - Pipeline can be synchronized with an external event, e.g. a clock
  - $N$ instructions can be synchronized to $M$ events
  - Pipeline metrics can be collected nonintrusively, e.g. instruction count
  - Speed can be reduced to mimic worst-case conditions
  - Enables advanced power management
  - Fully programmable, external, AMBA peripheral
## Comparing ARM Cores

<table>
<thead>
<tr>
<th>Feature</th>
<th>ARM996HS</th>
<th>ARM968E-S</th>
<th>ARM946E-S</th>
<th>ARM7TDMI-S</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM ISA</td>
<td>v5TE</td>
<td>v5TE</td>
<td>v5TE</td>
<td>v4T</td>
</tr>
<tr>
<td>Core logic</td>
<td>Clockless</td>
<td>Synchronous</td>
<td>Synchronous</td>
<td>Synchronous</td>
</tr>
<tr>
<td>Pipeline depth</td>
<td>5 stages</td>
<td>5 stages</td>
<td>5 stages</td>
<td>3 stages</td>
</tr>
<tr>
<td>Thumb-1</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>DSP extensions</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>Coprocessor I/F</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>HW divider</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>NMI</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Caches</td>
<td>✗</td>
<td>✗</td>
<td>4-128K / 4-128K</td>
<td>✗</td>
</tr>
<tr>
<td>TCM (I / D)</td>
<td>0-4MB / 0-4MB</td>
<td>0-4MB / 0-4MB</td>
<td>0-1MB / 0-1MB</td>
<td>✗</td>
</tr>
<tr>
<td>DMA</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Memory Management</td>
<td>Enhanced MPU</td>
<td>✗</td>
<td>MPU</td>
<td>✗</td>
</tr>
<tr>
<td>Main I/O bus</td>
<td>2 x AHB-Lite</td>
<td>1 x AHB-Lite</td>
<td>1 x AHB</td>
<td>1 x AHB</td>
</tr>
<tr>
<td>Bus architecture</td>
<td>Harvard</td>
<td>Von Neumann</td>
<td>Von Neumann</td>
<td>Von Neumann</td>
</tr>
</tbody>
</table>
# Power, Performance, Size

<table>
<thead>
<tr>
<th>Feature</th>
<th>ARM996HS</th>
<th>ARM968E-S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>50 Equivalent MHz (WC), 77 Equivalent MHz (NC)</td>
<td>100 MHz (WC)</td>
</tr>
<tr>
<td></td>
<td>54 DMIPS, 83 DMIPS (NC)</td>
<td>107 DMIPS (WC)</td>
</tr>
<tr>
<td>Area</td>
<td>&lt; 0.69 mm²</td>
<td>0.59 mm²</td>
</tr>
<tr>
<td></td>
<td>89 Kgates (nand2)</td>
<td>88 Kgates (nand2)</td>
</tr>
<tr>
<td>Power</td>
<td>0.045 mW/MHz</td>
<td>0.13 mW/MHz</td>
</tr>
</tbody>
</table>

- Numbers based on post-layout simulation
  - Artisan Sage-X 0.13µ TSMC process
  - WC = worst, 1.08 V, 125°C, NC = nominal, 1.2V, 25°C
- ARM968E-S netlist synthesized for 100 MHz
- **Power** simulations at 77 MHz under nominal conditions
- Equivalent MHz is the speed needed by an ARM968E-S to achieve the same performance as the ARM996HS
Noise and Electromagnetic Radiation in Digital Circuits

- Digital circuits generate
  - Voltage noise on power supply lines
    - Might disrupt operation of A/D converter drawing power from the same source
  - Induced currents in the silicon substrate
    - Might disrupt operation of A/D converter integrated on the same substrate

- Digital circuits emit
  - EM radiation at their clock frequency
  - EM radiation at higher harmonics of their clock frequency
    - Radio receiver might mistake these signals for radio signals
Supply Current: Time Domain

- In a clocked circuit, activity is maximal shortly after the productive clock edge and fades away with time.
- In terms of current:

  \[ I_{dd} \]

  \[ \text{time} \]

  ⇒ Noise in on-chip power and ground lines
- Local drops in supply voltage have impact
  - On performance
  - On circuit reliability
Low Current Peaks and Total Current

ARM996HS consumes \textbf{2.8x less power} than an ARM968E-S and \textit{reduces current peaks by a factor 2.4}

<table>
<thead>
<tr>
<th>Current (A)</th>
<th>Cumulative Energy (J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Current (A)</th>
<th>Cumulative Energy (J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Clock-gated ARM968E-S processor

Handshake ARM996HS processor
ARM996HS draws a relatively constant current, whereas the ARM968E-S current swings between 0 and 25-35 mA.
Current Peak Histogram

ARM996HS current typically between in 1 to 5 mA range, whereas ARM968E-S has significant current up to 20 mA.
Low Electromagnetic Emissions

ARM996HS offers low electromagnetic emissions across the whole radio spectrum

Clock-gated ARM968E-S processor

Handshake ARM996HS processor
ARM996HS Conclusions

- Fully ARMv5TE compatible
- Advanced features
  - Dual AHB-Lite interfaces
  - Enhanced MPU
  - Hardware divide
  - Nonmaskable interrupts
- Low gate count
- Low power
- Low current peaks
- Low electromagnetic emissions
- Zero (active) standby power
  - Zero (active) power wait-for-interrupt with immediate wake-up
- Robustness
ARM996HS Conclusions

- The processor is delivered as a firm core:
  - Targeted to customer’s standard-cell library
  - Hardening scripts including design constraints
  - Design-for-test (DfT) included
  - Full support for automatic test-pattern generation (ATPG)

- What the end customer does not need:
  - Special memories
  - Special cell libraries
  - Custom circuit design
  - Special process technology

- Core availability
  - Currently available for licensing

- Tools availability
  - Mature compiler and debugger support for ARM9E core
    - Compatible with ARM and third-party tool vendors
Thank you

For any questions please contact us via
http://www.handshakesolutions.com/Contact
Handshake Solutions
the first to offer a commercially viable way to exploit the benefits of self-timed circuits