Software Configurable Processors
Change System Design

Ricardo E. Gonzalez
Embedded System Design Dilemma

- COMPUTE
- PERFORMANCE
- SW
- HW
- SYSTEM TIME-TO-MARKET
- ASSP
- ASIC
- FPGA
- DSP
- MEDIA
- SECURITY
- ETC.
Solving Compute Intensive Problems

- Processor with embedded programmable logic
  - Lots of compute resources

- Utilize fabric by extending the ISA
  - Design your own FU
  - Add processor state

- Simple programming model
  - Issue instructions to perform computation

- Tailored for high-throughput applications
  - Compute intensive
Compute Intensive Markets & Applications

Military & Security
- Sonar/Radar
- Biometrics

Medical Electronics
- CAT Scan
- Ultrasound

Office Automation
- Printers
- Scanners

Networking
- Broadcast Equipment
- Audio Studio
- Encryption/Decryption
- Network Security

Video & Audio
- Broadcast Equipment
- Audio Studio
EEMBC Telemark

- Embedded Microprocessor Benchmark Consortium
- EEMBC creates, certifies and publishes performance benchmark results

OPTIMIZED S5000 Performance With ISEF Acceleration

OUT-OF-BOX

© 2005 Stretch Inc. All rights reserved.
Application Acceleration Process
**RGB → YC^bC^r**

**Color Conversion Function:**

```c
SE_FUNC /* Tells Stretch C-Compiler to reduce this function to an instruction */
Void RGB2YCBCR (WR A, WR *B) {
    char Y[4], Cb[4], Cr[4];
    char R[4], G[4], B[4];

    R[0] = A(23,16); G[0] = A(15,8); B[0] = A(7,0); /* and so on. . . */

    for (i = 0; i < 4; i++) {
        Y[i] = (77*R[i] + 150*G[i] + 29*B[i]) >> 8;
        Cb[i] = (32768 - 43*R[i] - 85*G[i] + (B[i] << 7)) >> 9;
        Cr[i] = (32768 + (R[i] << 7) - 107*G[i] - 21*B[i]) >> 9;
    }
    *B = (Y[3],Cr[3]+Cr[2],Y[2],Cb[3]+Cb[2],Y[1],Cr[1]+Cr[0],Y[0],Cb[1]+Cb[0]);
}
```

**Program Loop:**

```c
for (...) {
    WRGET0(&A, 12); /* Load 12 bytes (4 RGB pixels) */
    RGB2YCBCR(A, &B); /* Convert 4 pixels */
    WRPUT0(B, 8); /* Store 8 bytes (4 YCbCr pixels */
}
```
Programming Flow
Development & Debug

- Programmers develop & debug using familiar tools
- Faster debug cycle
- Port & accelerate apps within Stretch IDE
- No hardware design experience required!
Hardware Overview
S5 Datapath and Key Parameters

Wide Register File
- 32 128-bit entries

Load/store unit
- 128-bit load/store
- Auto increment/decrement
- Immediate, indirect, circular
- Variable-byte load/store
- Variable-bit load/store

ISEF
- 3 inputs and 2 outputs
- Pipelined, interlocked
- 32 16-bit MACs and 256 ALUs
- Bit-sliced for arbitrary bit-width
- Allow control logic
- Allow internal processor state

RISC Processor
- Tensilica – Xtensa V
- 32 KB I & D Cache
- On-Chip Memory, MMU
- 24 Channels of DMA, FPU
S5 Pipeline

- Standard 5-stage RISC pipeline
- Extension instruction may be much longer
  - Fully pipelined
- Compiler schedules all instructions
- Hardware interlocks ensure correctness
  - Instruction groups subdivided by use/def requirements
  - Interlocks driven by config table in Instruction Unit
Pipeline Schedule

- Extension Instructions may have different latencies
- Initiation interval of EI a function of:
  - ISEF clock ratio
  - WR and state use/def
- Statically scheduled
  - Pipeline behavior is completely known at compile time
- Optimal schedule typically has initiation interval > 1
  - No performance penalty for lower clock rate!
ISEF Architecture

- Array of 64-bit ALUs
  - May be broken at 4-bit or 1-bit boundaries
  - Conditional ALU operation: \( Y = C \ ? \ (A \ op_1 \ B) \ : \ (A \ op_2 \ B) \)
  - Registers to implement state, pipelining
- Array of 4x8 multipliers, cascadable to 32x32
  - Programmable pipelining
- Programmable routing fabric
- Execution clock divided from processor clock: 1:1, 1:2, 1:3, 1:4
- Up to 16 instructions per ISEF configuration
- Many configurations per executable
- Reconfiguration
  - User directed or on demand
  - <100 µsec for complete reconfiguration
Dynamic ISEF Configuration

Two ISEFs
- Each holds up to 16 instructions
- Configured via system bus
- Managed by DMA
- Independently configurable

On-Demand Configuration
- Auto managed by HW and OS
- Handled like cache miss

Configuration Preloading
- Managed by application
- Handled like instruction pre-fetch
Examples of Extension Instructions
Flexibility in FIR Implementations

8-MACs/Cycle

16-MACs/Cycle

32-MACs/Cycle

- S5000 Strengths
  - Up to 32-MAC/Cycle
  - Flexibility to trade-off ISEF usage vs. compute speed
  - Wide Load & Stores feed the compute array
Summary

- C/C++ Programmers can tailor the processor for their application
  - Continue to develop & debug in a familiar environment (IDE/gdb)

- Application-specific instruction provide 10-100X speedup
  - Modest porting effort

- C/C++ programmers can easily design high-performance systems
  - Enabling a new system design methodology