The CA1024:
A fully programmable system-on-chip for cost-effective HDTV media processing
Lazar Bivolarski, Bogdan Mitu, Anand Sheel, Gheorghe Stefan, Tom Thomson, Dan Tomescu
Connex Technology, Inc.

- Core asset: **ConnexArray™** an efficient data-parallel architecture
  - 200 MHz
  - 200 GOPS (16-bit simple integer operations)
  - 60 GOPS/Watt
  - 3.2 GB/sec external; 400 GB/sec internal

- Application domain: HDTV
Our Solution: Integral Parallel Machine

• Data-parallel computation: *ConnexArray*

• Time-parallel computation (supported by speculative parallelism): *Stream Accelerator*

• I/O process is transparent to the main data-parallel computational process: *I/OPlan & IOC*
The Connex Architecture

**Connex Array:**
1,024 linearly connected 16-bit Processing Cells

**Sequencer:**
32-bit stack machine & program memory & data memory issues in each cycle (on a 2-stage pipe) one 64-bit instruction for Connex Array and a 24-bit instruction for itself

**I/O Controller:**
32-bit stack machine controls a 3.2 GB/s I/O channel

**Processing Cell:**
Integer unit & data memory & Boolean unit

I/O channel works in parallel with code running on the Connex Array

---

Defining the Future of Video Processing
Connex Array Structure

- Processing Cells are **linearly connected** using only the register R0.
- **IO Plan** consists in all R1's supervised mainly by the IO Controller.
- **Conditional execution** based on the state of the Boolean unit.
- **Integer unit, Boolean unit and Data memory** execute in each cycle command fields from a 64-bit instruction issued by Sequencer.
- **Vector reduction operations** with scalar results in the TOS of Sequencer (receiving through a 3-stage pipe data from the array of cells).
Full Line Operations: Operate On All Elements in Parallel

```
Line i
Line j
Line k
```

16-bit data operand

```
+,-,*,XOR, etc.
```

Line k = Line i \ OP \ Line j

Line k = Line i \ OP \ scalar value (repeated for all elements)
Columns Active Based On Repeating Patterns

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1023</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line i</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line j</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line k</td>
<td></td>
<td></td>
</tr>
<tr>
<td>255</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example: Mark all odd columns active. Or mark every third column active. Or mark every third and fourth column active, etc.
Columns Active Based On Results of Previous Operations

Example: Apparently random columns are active, marked, based on Data-dependent results of previous operations. This enables selective processing based on data content.
Outer-Loop Parallelism:
Program in context of 128+ data-structure instances

Example: 8x8 DCT

<table>
<thead>
<tr>
<th>Line i</th>
<th>Line j</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Line i</td>
<td>Line j</td>
</tr>
<tr>
<td>255</td>
<td>255</td>
</tr>
</tbody>
</table>

Example: 128 sets of 8x8 run in parallel in a 1024-cell array
The Fine-Grain Parallelism allows different algorithms to be applied at the same time for increased parallelism.
Local Memory Mapping Based on Data Dependency

Local data dependency remapping and processing of multiple neighboring blocks enables high degree of parallelism.
Programming Connex

- **CPL** (Connex Programming Language) is an extension of C
- Code that operates on scalar data written in regular C notation
- Connex-specific operators defined for features not available in C, e.g. operations on vectors, selections
- CPL uses sequential operators and control structures on vector and select data-types
- Using CPL the Connex Machine is programmed the same way as conventional sequential machines

```c
{ ... 
  const short OFFSET = 15;
  ...
  short vector x, y;
  short vector min, max;
  ...
  sel = all;
  x += OFFSET;
  ...
  min = x;
  max = x;
  min = (min > y)? y; /* min = min(x, y) */
  max = (max < y)? y; /* max = max(x, y) */
  ...
}

Vectors are arrays of scalar components.

Selections are arrays of Boolean values that dictate what vector components are active.
The main strategic decisions in defining Connex Architecture

- **Simple** architecture:
  - nothing spectacular at the circuit level
  - no technological challenges

- **Fully programmable** (no pieces of hardware to solve critical problems)

- **Tuned** on the application domain (HDTV)

- **Programming language** able to hide the structural details (because they are simple)
  - Efficient compiler
  - Cycle accurate simulator

- **Imaginative algorithms** to adapt the architecture to the application domain
What differentiate Connex from other Parallel Architectures

• All forms of parallelism are strongly segregated
  – **ConnexArray** for data-parallel computation
  – **Stream Accelerator** for time-parallel (speculative) computation

• The granularity perfectly fits the application domain
  – 16-bit **small & simple** processing elements
  – enough local data memory (256 16-bit words)
  – **no** MACs, **no** FPUs, **no** multipliers…

• The simplest interconnection network allowed by the **parallel computational locality**

• “Smart” IO process able to save computation or supported by additional computation for IO bounded applications
Performances

- > 2 GOPS/mm$^2$ (peak performance)
- 60 GOPS/Watt
- Dot Product: 28 cycles (16-bit 1Kcomponent vectors)
- DCT: 0.35 clock cycle per pixel
- SAD: 0.0025 clock cycle per pixel
- Using 83% of ConnexArray computational power decodes H.264 dual HD stream
## Performance Comparisons

### 16-bit Fixed-Point Sum of Absolute Differences (16x16 SAD - Motion Estimation)

<table>
<thead>
<tr>
<th>Processor</th>
<th>SAD/MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog Devices BF651</td>
<td>100X</td>
</tr>
<tr>
<td>TI C64xx</td>
<td>50X</td>
</tr>
<tr>
<td>Equator BSP16-500</td>
<td>25X</td>
</tr>
<tr>
<td>Connex CA1024</td>
<td>100X</td>
</tr>
</tbody>
</table>

### 16-bit Fixed-Point Discrete Cosine Transform (8x8 DCT - Image Compression)

<table>
<thead>
<tr>
<th>Processor</th>
<th>DCT/MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog Devices BF651</td>
<td>100X</td>
</tr>
<tr>
<td>Tensilica VectraDSP</td>
<td>70X</td>
</tr>
<tr>
<td>TI C64xx</td>
<td>20X</td>
</tr>
<tr>
<td>Connex CA1024</td>
<td>100X</td>
</tr>
</tbody>
</table>
### ConnexArray Performance Decoder

**VC-1 Dual HD Stream**

<table>
<thead>
<tr>
<th>Step</th>
<th>Clock Cycles/ Macro-Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dez zigzagging</td>
<td>24.7</td>
</tr>
<tr>
<td>AC Prediction</td>
<td>23.3</td>
</tr>
<tr>
<td>DC Prediction</td>
<td>16.3</td>
</tr>
<tr>
<td>IT/IQ</td>
<td>106.7</td>
</tr>
<tr>
<td>Overlap Transform</td>
<td>20.8</td>
</tr>
<tr>
<td>Motion Vector Reconstruction</td>
<td>20</td>
</tr>
<tr>
<td>Motion Vector Compensation</td>
<td>35.3</td>
</tr>
<tr>
<td>Loop Filter</td>
<td>15.4</td>
</tr>
<tr>
<td>Deringing Filter</td>
<td>14.3</td>
</tr>
<tr>
<td><strong>Total</strong> [ Clock cycles/ macro-block ]</td>
<td><strong>276.8 (67%)</strong></td>
</tr>
</tbody>
</table>

**Allowed Clock cycles/macro-block (2 channel, 1080i): 409 Clocks/MB**
CA1024 Project Status

- TSMC 0.13 micron
- 200 MHz clock rate
- Standard ASIC flow
- 676-pin PBGA
- Samples Q4 2006
Thank You!
Back-up slides
Connex Value Proposition

• **Fully programmable** solution for HDTV video encoding, decoding, trans-coding and post-processing

• **Silicon efficient** architecture with die size competitive with similar function ASICs

• **High performance** to enabling multi-standard, multi-channel HDTV
### ConnexArray Performance Decoder

**H.264 Dual HD Stream**

<table>
<thead>
<tr>
<th></th>
<th>Clock Cycles/Macroblock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dezigzagging</td>
<td>37.3</td>
</tr>
<tr>
<td>Intra Prediction</td>
<td>54.1</td>
</tr>
<tr>
<td>IT/IQ</td>
<td>97.3</td>
</tr>
<tr>
<td>Motion Compensation</td>
<td>114.3</td>
</tr>
<tr>
<td>Deblocking Filter</td>
<td>27.1</td>
</tr>
<tr>
<td><strong>Total</strong> [ Clock Cycles/Macroblock ]</td>
<td><strong>337.8</strong></td>
</tr>
</tbody>
</table>

**Allowed Clock cycles/Macroblock (2 channel, 1080i): 409 Clks/MB**
StreamAccelerator performing H.264 CABAC Decoding

- Targeted profile and level: 4.1 Main Profile
- Bit-rate/stream considered: 25Mbps
- Number of bins to decode using CABAC: 35M/sec
- Number of clock cycles per bin: < 2 cycles
- Cycles to decode bins/stream: 70M
- Typical bit-rate expected for DVB: 10Mbps
- Cycles to decode bins for typical stream (DVB): 30M
- Available cycles/stream: 100M
Device Cost Comparison

Relative Pad Limited Die Size

Assumptions:
1) Die Size is pad limited
2) Staggered, minimum pitch pads
3) All devices are in 130nm process