DAPDNA-2
A Dynamically Reconfigurable Processor with 376 32-bit Processing Elements

Tomoyoshi Sato
Vice President & CTO
IPFlex Inc.
Agenda

- Overview
- Design Goals and Decisions
- Overall Architecture
- Processing Element (PE) Architecture
- Interconnect Architecture
- Application Construction
- Performance
- Advanced Usages
- Summary
DAPDNA-2

- 32bit RISC + Reconfigurable Fabric + Peripherals
- Fujitsu 0.11µm 7Cu+1Al
- 12 M gates
- 1156-pin FCBGA, 2.4V I/O, 1.2V Core
- 166 MHz, 3-7 W

- Suited for stream processing
- 10-50x performance of 3GHz general-purpose CPU
Design Goals and Decisions

- **High performance**
  - Massively parallel processing elements
- **Flexibility**
  - Field programmable
- **Versatility**
  - Dynamically reconfigurable with small overhead
- **Ease of use**
  - Fixed-frequency coarse-grained ALUs
- **Scalability**
  - High-bandwidth I/O (interconnect) channels
DAPDNA-2 Architecture

- **CPU for sequential tasks**
  - DAP (Digital Application Processor)
  - 32-bit RISC with 2way 8k+8k I/D caches

- **Reconfigurable fabric for parallel processing**
  - DNA (Distributed Network Architecture)
    - 376 heterogeneous 32-bit processing elements
    - ALUs, RAMs, delays, counters, I/O buffers
    - 4 configuration banks: switchable in one cycle
    - 28 billion ALU operations / s (166MHz x 168)
    - 9 billion 16x16 multiplications / s (166MHz x 56)
  - 2.66 GB/s memory bandwidth (64-bit DDR166)
  - 4 GB/s I/O bandwidth (32-bit x 166MHz x 6ch)
DAPDNA-2 Block Diagram
DAPDNA-2 Reconfigurable Fabric

Segment 0

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDB</td>
<td>LDB</td>
<td>LDB</td>
<td>LDB</td>
<td>LDB</td>
<td>EXC</td>
<td>EXC</td>
<td>EXF</td>
</tr>
<tr>
<td>RAM</td>
<td>C16L</td>
<td>C16L</td>
<td>C16L</td>
<td>C16E</td>
<td>C16E</td>
<td>C16E</td>
<td>C16E</td>
</tr>
<tr>
<td>RAM</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
</tr>
<tr>
<td>RAM</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
</tr>
<tr>
<td>RAM</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
</tr>
<tr>
<td>RAM</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
</tr>
<tr>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
</tr>
<tr>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
</tr>
</tbody>
</table>

Segment 1

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
</tr>
<tr>
<td>RAM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
</tr>
<tr>
<td>RAM</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
</tr>
<tr>
<td>RAM</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
</tr>
<tr>
<td>RAM</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
</tr>
<tr>
<td>RAM</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
</tr>
<tr>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
</tr>
<tr>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
</tr>
</tbody>
</table>

Segment 2

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
</tr>
<tr>
<td>RAM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
</tr>
<tr>
<td>RAM</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
</tr>
<tr>
<td>RAM</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
</tr>
<tr>
<td>RAM</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
</tr>
<tr>
<td>RAM</td>
<td>C16S</td>
<td>C16S</td>
<td>C16S</td>
<td>C16E</td>
<td>C16E</td>
<td>C16E</td>
<td>C16E</td>
</tr>
<tr>
<td>STB</td>
<td>STB</td>
<td>STB</td>
<td>STB</td>
<td>STB</td>
<td>STB</td>
<td>STB</td>
<td>STB</td>
</tr>
</tbody>
</table>

Segment 3

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDX</td>
<td>LDX</td>
<td>LDX</td>
<td>LDX</td>
<td>LDX</td>
<td>LDX</td>
<td>LDX</td>
<td>LDX</td>
</tr>
<tr>
<td>DLH</td>
<td>C32L</td>
<td>C32L</td>
<td>C32L</td>
<td>C32L</td>
<td>C32E</td>
<td>C32E</td>
<td>C32E</td>
</tr>
<tr>
<td>DLH</td>
<td>EXF</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
</tr>
<tr>
<td>DLH</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
</tr>
<tr>
<td>DLH</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
</tr>
<tr>
<td>DLH</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
</tr>
<tr>
<td>DLH</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
</tr>
<tr>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
</tr>
</tbody>
</table>

Segment 4

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLX</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
</tr>
<tr>
<td>DLH</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
</tr>
<tr>
<td>DLH</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
</tr>
<tr>
<td>DLH</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
</tr>
<tr>
<td>DLH</td>
<td>EXS</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
<td>EXC</td>
</tr>
<tr>
<td>DLH</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
</tr>
<tr>
<td>DLH</td>
<td>EXM</td>
<td>EXS</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
<td>EXR</td>
</tr>
<tr>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
</tr>
</tbody>
</table>

Segment 5

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLX</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
<td>DLV</td>
</tr>
<tr>
<td>DLH</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
<td>EXM</td>
</tr>
<tr>
<td>DLH</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
<td>DLE</td>
</tr>
<tr>
<td>DLH</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
<td>EXF</td>
</tr>
<tr>
<td>DLH</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
<td>EXS</td>
</tr>
<tr>
<td>DLH</td>
<td>C32S</td>
<td>C32S</td>
<td>C32S</td>
<td>C32S</td>
<td>C32E</td>
<td>C32E</td>
<td>C32E</td>
</tr>
<tr>
<td>STX</td>
<td>STX</td>
<td>STX</td>
<td>STX</td>
<td>STX</td>
<td>STX</td>
<td>STX</td>
<td>STX</td>
</tr>
</tbody>
</table>

HotChips 17 16 Aug 2005
### Processing Elements

<table>
<thead>
<tr>
<th>Type</th>
<th>Qty.</th>
<th>Description / Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>168</td>
<td>EXM : ALU with 16x16 -&gt; 32 multiplier</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EXS : ALU with byte-swap instruction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EXC, EXR, EXF :</td>
</tr>
<tr>
<td>RAM</td>
<td>32</td>
<td>16kB, 8/16/32-bit data width</td>
</tr>
<tr>
<td>delay</td>
<td>136</td>
<td>DLE : programmable (1-13) delay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DLV : vertical cross-segment delay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DLH, DLX :</td>
</tr>
<tr>
<td>counter</td>
<td>24</td>
<td>C16L : 16bit address counter for load</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C16S, C16E, C32L, C32S, C32E :</td>
</tr>
<tr>
<td>I/O buffer</td>
<td>16</td>
<td>LDB : load buffer from external memory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDX : load buffer from I/O channel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STB, STX :</td>
</tr>
</tbody>
</table>
PE Example: ALU with 16x16 MUL

- Finely pipelined
  - Guarantees 166-MHz operation
- Registered input/output
  - Gives sufficient time to interconnect wires
- ALU feedback path
  - Supports accumulation in one-cycle throughput
- Stage bypass path
  - Reduces latency when some stages are unused
Segmented Interconnect

- 8x8 PEs in 1 segment
- 1-cycle intra-segment interconnect
  - Nearly freely routable
- Inter-segment routing via delay elements on segment boundaries
  - Nearest-neighbor connection
Intra-Segment Interconnect Concept

- H-Bus
- V-Bus
- H-to-V MUX
- V-to-PE MUX
Intra-Segment Routing Example

- Considering single net
  - 100% routable from arbitrary source to arbitrary destinations

- Considering multiple nets (real application)
  - Contentions may occur
  - Need careful placement to avoid contentions
Intra-Segment Routing Contention

Unroutable if

$$net1\.src\.row == net2\.src\.row$$

$$net1\.dst\.col == net2\.dst\.col$$

Too restrictive

V-BUS Contention
“At most one signal from a row” to each column-L (or column-R)

“At most two signals from two rows” to each column-L (or column-R)
## Intra-Segment Interconnect Comparison

<table>
<thead>
<tr>
<th>Type</th>
<th>MUX structure</th>
<th>MUX Qty. (*1)</th>
<th>Routing delay (*2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal full-crossbar</td>
<td>64-to-1 x 128 (PE-to-PE)</td>
<td>8064</td>
<td>7 + 6</td>
</tr>
<tr>
<td>Rev.1</td>
<td>8-to-1 x 128 (H-to-V)</td>
<td>1792</td>
<td>7 + 6</td>
</tr>
<tr>
<td></td>
<td>8-to-1 x 128 (V-to-PE)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rev.2</td>
<td>16-to-1 x 128 (H-to-V)</td>
<td>2816</td>
<td>8 + 7</td>
</tr>
<tr>
<td></td>
<td>8-to-1 x 128 (V-to-PE)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(*1) number of underlying 2-to-1 32-bit MUXs  
(*2) levels of FO2 buffers + 2-to-1 MUXs
Inter-Segment Routing

Each cross-segment delay PE actually has two inputs and two outputs to provide enough bandwidth.
Application Example: 4-Tap FIR Filter

Schematic

PE NetList

Config Data

$x[i]$ → $h0$ → $h1$ → $h2$ → $h3$ → $y[i]$ 

166Msample/s

Address Counter → Load Buffer → $h0$ → $h1$ → $h2$ → $h3$ → Store Buffer

Retime & Tech-map

Place & Route

B0B7ECC7
AE10CB59
B3392B7D
711073AF
EEE10791
C211A00B
48101D03
2871D37B
579D389C
DE771107
4C046D5D
EEE0AD06
97B655F3
3DED0F10
25160D93
5599F0FC
8AC3B0BD

HotChips 17 16 Aug 2005
Alternative: C-Based Design

Source Code Editor/ Debugger

NetList Viewer

Breakpoint
## Performance Comparison

<table>
<thead>
<tr>
<th>Application</th>
<th>Pentium4 3 GHz</th>
<th>DAPDNA-2 166 MHz</th>
<th>Performance Multiple</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024pt FFT (k iterations/s)</td>
<td>8.2</td>
<td>325</td>
<td>40</td>
</tr>
<tr>
<td>3x3 Average Filter (M pixels/s)</td>
<td>15.5</td>
<td>666</td>
<td>43</td>
</tr>
<tr>
<td>7x7 FIR Filter (M pixels/s)</td>
<td>3.0</td>
<td>166</td>
<td>55</td>
</tr>
<tr>
<td>Binary Pixel Expansion / Contraction (M pixels/s)</td>
<td>32.7</td>
<td>666</td>
<td>20</td>
</tr>
<tr>
<td>Floyd-Steinberg Error Diffusion (M pixels/s)</td>
<td>11</td>
<td>304</td>
<td>28</td>
</tr>
<tr>
<td>Jarvis-Judice-Ninke Error Diffusion (M pixels/s)</td>
<td>6.2</td>
<td>110</td>
<td>18</td>
</tr>
</tbody>
</table>

*Simulation data
Advanced Usage: Dynamic Reconfiguration

- Exploits “temporal computing”
- 4 banks of configuration memory
  (1 foreground bank + 3 background banks)
- Can copy from a background bank
to the foreground bank (6 kB) in one cycle
Advanced Usage: Multiple Chips

- Exploits “spatial computing”
- Six 32-bit I/O channels for inter-chip interconnect
- I/O channels can be treated as delay elements

Example: MPEG-2 realtime encoder using 2 chips

Format : MP@HL
Resolution : 1920x1080
Frame rate: 30 fps
Latency : 46 ms
DAPDNA-2 Summary

- High performance
  - thanks to massively parallel processing elements
- Easy to use
  - thanks to fixed-frequency coarse-grained ALUs
- Dynamically reconfigurable in one cycle
- Scalable via high-bandwidth I/O channels
- Schematic-based or C-based application design flow
http://www.ipflex.com/