C-Based Hardware Design Platform for Dynamically Reconfigurable Processor

September 22nd, 2005

IPFlex Inc.
Agenda

- Merits of C-Based hardware design
- Hardware enabling C-Based hardware design
- DAPDNA-FW II Design Tool
- Programming Example and Performance
Short development iteration while achieving high performance

Until Now…

Development Idea
→ Algorithm & System Design
→ Logic Design
→ Physical Design
→ Test Chip
→ Product

Long re-design iterations take 80% of system development time

IPFlex’s Technology Enables…

Development Idea
→ Algorithm & System Design
→ SW Design = HW Design

Reduced Development Cycle = Reduced Risk

Direct and Timely Link between Idea and Product
HW and SW enabling short iteration cycle

**Dynamically Reconfigurable Hardware**
Hardware functions reconfigured in a single clock

**Software to Silicon™**
Hardware design with C-like language
Agenda

- Merits of C-Based hardware design
- Hardware enabling C-Based hardware design
- DAPDNA-FW II Design Tool
- Programming Example and Performance
RISC core and dynamically reconfigurable data flow machine

- 166MHz
- 376 32 bit processing elements
- 512KB Internal RAM
- Direct I/O
- DDR SDRAM
- Processor chip, design tool, evaluation boards available
RISC core and dynamically reconfigurable data flow machine

PE Matrix

DNA

PE Matrix (376 PEs)

Configuration Memory

DNA Direct I/O

Hi Speed Switching BUS

Peripheral

S-ROM
UART
GPIO
INTC
Timer

DAP
Debug I/F

RISC Core

Instruction Cache
Data Cache

PCI I/F
DDR-SDRAM I/F
DMA Controller
DAP RISC and DNA Parallel Data Processing

```
.L8:    cmp g e t 3 , 8  
br/ t . L 1 5
.L9:    mov          t8,cos_table1  
mov          t3,cos_table2  
addv         t2,t8,t2  
ld           t7,t3,3  
ld           t6,[fp+4]  
addv         t2,t6,t2  
li           t7,t7,1  
:            :            :
addv         t8,t8,11  
addv         t7,t7,10  
addv         t4,t4,a1  
.L12:     mov          t9,1  
addv         t2,t2,19  
br           .L10  
.L13:     .L14:     mov          t2,1  
addv         t2,t2,12  
br           .L8c
```

Memory

REG

IDU

EU

PE matrix

Config Mem

Memory
<table>
<thead>
<tr>
<th>PE</th>
<th>#</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXE</td>
<td>112</td>
<td>• 32 bit 2 inputs / 1 output execution elements</td>
</tr>
<tr>
<td>EXM</td>
<td>56</td>
<td>• 16bitx16bit-&gt;32bit multipliers</td>
</tr>
<tr>
<td>DLE</td>
<td>136</td>
<td>• 32 bit 2 inputs / 1 output delay element</td>
</tr>
<tr>
<td>RAM</td>
<td>32</td>
<td>• DNA Internal memory element</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 16KB×32 =512KB</td>
</tr>
<tr>
<td>C16E</td>
<td>24</td>
<td>• Address generators</td>
</tr>
<tr>
<td>C32E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDx</td>
<td>16</td>
<td>• I/O buffers</td>
</tr>
<tr>
<td>STx</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Total**: 376
Scalability with DNA Direct I/O

16 DNAs at 16Gbps flow-thru (Total 32 Gbps I/O)

- Synchronize with the external clock
- 166MHz, 96 bit
- Interconnection between DNAs equivalent to DLE between segments
DAPDNA architecture: solution at any system size

System Example

4 Pipelined Processes
Dynamic Reconfiguration
Scalability

Small scale
Mid/Large size
Ultra large

One DNA Plane
One DAPDNA Processor
Up to 16 DAPDNA Processors

©2005 IPFlex Inc. All rights reserved.
Agenda

- Merits of C-Based hardware design
- Hardware enabling C-Based hardware design
- **DAPDNA-FW II Design Tool**
- Programming Example and Performance
Data Flow C compiler: A part of Software to Silicon design tool

System Specifications
Algorithm Design
C Source Code
Profiler
Block Diagram

DNA Compiler
(Dynamic Reconfiguration, Place & Route)

DAP DNA
DNA P.A.

C for DAP
Partitioning
Wrapper for DNA Configuration

DAP Compiler
DAPDNA Co-Simulator / Emulator

DNA Configuration

C for DNA
DFC Compiler
DNA Designer
DNA Blockset

MATLAB/Simulink

: Tools included in DAPDNA-FW II

©2005 IPFlex Inc. All rights reserved.
DFC Compiler: HW configurations from C language

C based Data Flow Description

DNA Configuration

DNA Control code for DAP

DFC Compiler
HW configurations written in DFC as functions

Application (DAP C code)

```c
void top_level()
{
  ...
  fir(d,h,r);
  ...
}
```

```c
void fir(int d[], int h[], int r[])
{
  dna_cfgram_load3(BANK3, cfg_fir);
  dna_config(0,BANK3);
  dna_run();
  ...
  dna_stop();
  ...
}
```

DNA Control Code

DNA Hardware

- Load config data
- Bank Switch
- DNA Run
- DNA Stop
HW configurations switched dynamically in single clock (6 nano second)

Application (DAP C code)

```c
... 
while (1) {
    ...
    VLC_Decoder(Vector, Quant_dat, Input);
    Inv_Quantize(dct_dat, quant_dat);
    Inv_DCT(mc_dat, dct_dat);
    Motion_Comp(mcu_dat, mcu_dat);
    Color_Conv(mcu_dat, mcu_dat);
    ...
}
```
Extension on ANSI C for data flow systems

- **Delayline: delayed data signals**

```c
delayline d;
d = a;
```

- **SEQ: Static code replication**

```c
int h[] = {1,2,3};
int z[];
seq (i=0; i<3; i++) {
    z[i] = a * h[i];
}
```
Advantages of DAPDNA architecture

- **Short development period**
  - Rapid prototyping
  - Co-Simulation between Processor (DAP RISC) and Data Flow Machine (DNA PE Matrix)
  - Deterministic placement / guaranteed clock speed @ 166Mhz

- **Flexibility to change HW according to the application in demand**
  - Dynamic reconfiguration in single clock

- **Performance close to custom device**
  - Deep pipelining and massive parallelization
  - 376 processing elements
Agenda

- Merits of C-Based hardware design
- Hardware enabling C-Based hardware design
- DAPDNA-FW II Design Tool
- Programming Example and Performance
Using dynamic reconfigurability to change image filters

Various image filters written in DFC and compiled

- Laplacian
- Binary
- Average
- Etc.

Filters run on DAPDNA-2 using dynamic reconfigurability of hardware
DAP calls DNA configurations with simple function calls

List of filters written in Data Flow c

An Average3x3 filter being called in main.c as a function

Main.c for DAP RISC
DFC-described filters compiled into HW with a single click of build button…

Average 3x3 filter described in DFC uses delayline and seq commands to take advantage of parallelization.
...resulting in a logical representation of the filter with 32-bit processing elements...
...as well as physical mapping on DNA
# DFC compiler result for image filters

<table>
<thead>
<tr>
<th>Image Filters</th>
<th>DAPDNA-2 (166 Mhz)</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EXE</td>
<td>EXM</td>
</tr>
<tr>
<td>3x3 Average</td>
<td>Usage %</td>
<td>12.5</td>
</tr>
<tr>
<td>Binary</td>
<td>Usage %</td>
<td>5.4</td>
</tr>
<tr>
<td>Binary Image Edge Det.</td>
<td>Usage %</td>
<td>5.4</td>
</tr>
</tbody>
</table>
Optimized result with DNA Designer
(Block Diagram) tool

<table>
<thead>
<tr>
<th>Image Filters</th>
<th>Images</th>
<th>Performance (M Pixels / s)</th>
<th>Performance Multiple</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Pentium 4* (3.06Ghz)</td>
<td>DAPDNA-2 (166Mhz)</td>
</tr>
<tr>
<td>3x3 Average</td>
<td></td>
<td>15.5</td>
<td>664.0</td>
</tr>
<tr>
<td>3x3 Laplacian (Edge Enhancement)</td>
<td></td>
<td>15.8</td>
<td>664.0</td>
</tr>
<tr>
<td>3x3 Laplacian (Edge Detection)</td>
<td></td>
<td>15.8</td>
<td>664.0</td>
</tr>
<tr>
<td>Binary</td>
<td></td>
<td>225.0</td>
<td>839.0</td>
</tr>
<tr>
<td>Binary Image Edge Detection</td>
<td></td>
<td>38.6</td>
<td>664.0</td>
</tr>
</tbody>
</table>

* Pentium4@3.06GHz, Red Hat Linux 8.0, gcc 3.2, SIMD instruction not used, 24bit color, 800x600 pixels, 1000 Repetitions, Average performance (pixel/sec)
**FFT: Protein structure analysis**

- Extrapolating molecular structure from X-ray diffraction image

- Extrapolation Algorithm (HIO)

  - **Reciprocal Space**
  - **Real Space**

- Requires 1024 X 1024 32-bit FFTs

Source: RIKEN
DAPDNA-2 is 10X faster than P4 with 1/200 energy

* Assuming 80W for P4 and 4W for DAPDNA-2
Source: IPFlex experimentation

©2005IPFlex Inc. All rights reserved.
10Gbps active firewall by NTT

Packet processing at 10Gbps wirespeed
“Hot” updates

Protects network

High-speed gateway link

High-speed server link

Protects access line

10Gb/s Firewall System using Reconfigurable Processors
Reconfigurable system technical committee

©2005IPFlex Inc. All rights reserved.
10Gbps active firewall by NTT

- **“Hot” update capability**
  - Hardware circuits can be updated without interrupting service

- **Guaranteed 10Gbps packet processing**
  - 10Gbps wire-speed is guaranteed, even with a succession of short packets

- **L4 flow classification**
  - Various flow classifications supported, including inside and outside of Netmask
IPFlex’s technology / product roadmap

General MPU like Pentium and DNA+AXION complement well

<table>
<thead>
<tr>
<th>Granularity of operands</th>
<th>Coarse</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>General MPU</td>
<td>AXION</td>
</tr>
<tr>
<td>High</td>
<td>DNA</td>
<td></td>
</tr>
</tbody>
</table>

Realizes extreme performance

<table>
<thead>
<tr>
<th>Performance</th>
<th>2005</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>IPFlex-enhanced multi-core Processor series</td>
</tr>
<tr>
<td></td>
<td>DAPDNA-2 “Super chip”</td>
</tr>
<tr>
<td>Low</td>
<td>Intel multi-core Pentium series</td>
</tr>
<tr>
<td></td>
<td>Pentium Extreme edition</td>
</tr>
<tr>
<td></td>
<td>Pentium series</td>
</tr>
<tr>
<td></td>
<td>Single-core</td>
</tr>
<tr>
<td></td>
<td>Multi-core</td>
</tr>
</tbody>
</table>

Degree of computational parallelism
Dynamically Reconfigurable Processor
and
C-Based Hardware Design Platform

by

IPFlex Inc.

www.ipflex.com
info@ipflex.com