The Raw Tiled Processor Architecture

Is the future of architecture in tiles?

Anant Agarwal
MIT

http://www.cag.csail.mit.edu/raw
A tiled processor architecture prototype: the Raw microprocessor
Embedded system:
1020 Element Microphone Array
1020 Node Beamformer Demo

- 2 People moving about and talking
- Track movement of people using camera (vision group) and display on monitor
- Beamformer focuses on speech of one person
- Select another person using mouse
- Beamformer switches focus to the speech of the other person
The opportunity

20MIPS cpu
1987
The opportunity

2007

The billion transistor chip
Enables seeking out new application domains

- Redefine our notion of a “general purpose processor”

- Imagine a single-chip handheld that is a speech driven cellphone, camera, PDA, MP3 player, video engine

- Imagine a single-chip PC that is also a 10G router, wireless access point, graphics engine

- While running the gamut of existing desktop binaries
But, where is general purpose computing today?
How does the ASIC do it?

- Lots of ALUs, lots of registers, lots of small memories
- Hand-routed, short wires
- Lower power (everything close by)
- Stream data model for high throughput

But, not general purpose
Our challenge

How to exploit ASIC-like features
Lots of resources like ALUs and memories
Application-specific routing of short wires

While being “general purpose”
Programmable
And even running ILP-based sequential programs

One Approach: Tiled Processor Architecture (TPA)
Tiled Processor Architecture (TPA)

- Lots of ALUs and regs
- Short, prog. wires
- Lower power

Programmable. Supports ILP and Streams
A Prototype TPA: The Raw Microprocessor

[Billion transistor IEEE Computer Issue '97]

A Raw Tile

The Raw Chip

Software-scheduled interconnects (can use static or dynamic routing - but compiler determines instruction placement and routes)
Tight integration of interconnect

The Raw Chip

Packet stream
Disk stream
Video1
RDRAM

Point-to-point bypass-integrated compiler-orchestrated on-chip networks
How to “program the wires”

Tile 10

```plaintext
fmul r24, r3, r4
```

route P→E

software controlled crossbar

Tile 11

```plaintext
fadd r5, r3, r25
```

route W→P

software controlled crossbar
The result of orchestrating the wires

C program
ILP computation

MPI program

httpd  Zzzz

Custom Datapath Pipeline
We have replaced
Bypass paths, ALU-reg bus, FPU-Int. bus,
reg-cache-bus, cache-mem bus, etc.

With a general, point-to-point, routed
interconnect called:

Scalar operand network (SON)
Fundamentally new kind of network
optimized for both scalar and
stream transport
Programming models and software for tiled processor architectures

- Conventional scalar programs (C, C++, Java)
  Or, how to do ILP

- Stream programs
Scalar (ILP) program mapping

E.g., Start with a C program, and several transformations later:

```c
v2.4 = v2
seed.0 = seed
v1.2 = v1
pval1 = seed.0 * 3.0
pval0 = pval1 + 2.0
tmp0.1 = pval0 / 2.0
pval2 = seed.0 * v1.2
tmp1.3 = pval2 + 2.0
pval3 = seed.0 * v2.4
tmp2.5 = pval3 + 2.0
pval5 = seed.0 * v2.4
tmp3.6 = pval5 + 2.0
pval6 = tmp1.3 - tmp2.5
v2.7 = pval6 * 5.0
v2.7 = pval6 * 5.0
v1.8 = tmp1.3 - tmp2.5
v1.8 = tmp1.3 - tmp2.5
v0.9 = tmp0.1 - v1.8
v3.10 = tmp3.6 - v2.7
tmp2 = tmp3.6 - v2.7
v1 = v1.8;
tmp1 = tmp1.3
v0 = v0.9
tmp0 = tmp0.1
v3 = v3.10
v3 = v3.10
tmp3 = tmp3.6
v2 = v2.7
```
Scalar program mapping

\[
\begin{align*}
v2.4 &= v2 \\
seed.0 &= seed \\
v1.2 &= v1 \\
pval1 &= seed.0 \times 3.0 \\
pval0 &= pval1 + 2.0 \\
tmp0.1 &= pval0 / 2.0 \\
pval2 &= seed.0 \times v1.2 \\
tmp2.5 &= pval2 + 2.0 \\
pval3 &= seed.0 \times v2.4 \\
tmp3.6 &= pval3 / 3.0 \\
pval5 &= seed.0 \times 6.0 \\
pval4 &= pval5 + 2.0 \\
tmp3 &= tmp3.6 \\
v3.10 &= tmp3.6 - v2.7 \\
v3 &= v3.10 \\
tmp0 &= tmp0.1 \\
v0 &= v0.9 \\
v2 &= v2.7 \\
\end{align*}
\]
Placement

Tile1

Tile2

Tile3

Tile4
Routing

Processor code

Switch code

Tile1

Tile2

Tile3

Tile4
class BeamFormer extends Pipeline {
    void init(numChannels, numBeams) {
        add(new SplitJoin() {
            void init() {
                setSplitter(Duplicate());
                for (int i=0; i<numChannels; i++) {
                    add(new FIR1(N1));
                    add(new FIR2(N2));
                }
                setJoiner(RoundRobin());
            }
        });
        add(new SplitJoin() {
            void init() {
                setSplitter(Duplicate());
                for (int i=0; i<numBeams; i++) {
                    add(new VectorMult());
                    add(new FIR3(N3));
                    add(new Magnitude());
                    add(new Detect());
                }
                setJoiner(Null());
            }
        });
    }
}
Raw Beamformer Layout (by hand)
.18 micron process, 16 tiles, 425MHz, 18 Watts (vpenta)  
Of course, custom IC designed by industrial design team could do much better
Raw motherboard
More Experimental Systems

Systems Online or in Pipeline

- Raw Workstation
- Raw-based 1020 Microphone Array
- Raw 802.11a/g wireless system (collab with Engim)
- Raw Gigabit IP router
- Raw graphics system
- Raw supercomputing fabric
# Empirical Evaluation

Compare to P3 implemented in similar technology

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Raw (IBM ASIC)</th>
<th>P3 (Intel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Litho</td>
<td>180 nm</td>
<td>180 nm</td>
</tr>
<tr>
<td>Process</td>
<td>CMOS 7SF</td>
<td>P858</td>
</tr>
<tr>
<td>Metal Layers</td>
<td>Cu 6</td>
<td>Al 6</td>
</tr>
<tr>
<td>FO1 Delay</td>
<td>23 ps</td>
<td>11 ps</td>
</tr>
<tr>
<td>Dielectric $k$</td>
<td>4.1</td>
<td>3.55</td>
</tr>
<tr>
<td>Design Style</td>
<td>Standard Cell</td>
<td>Full custom</td>
</tr>
<tr>
<td></td>
<td>SA27E ASIC</td>
<td></td>
</tr>
<tr>
<td>Initial Freq</td>
<td>425 MHz</td>
<td>500-733 MHz (use 600)</td>
</tr>
<tr>
<td>Die Area</td>
<td>331 mm$^2$</td>
<td>106 mm$^2$</td>
</tr>
</tbody>
</table>

Raw #s from cycle-accurate simulator validated against real chip
-- FPGA mem controller in Raw
-- Raw SW i-caching adds 0-30% ovhd
Performance Results

~10x parallelism
~ 4x ld/store elim
~ 4x stream mem bw
Raw IP Router

Gb/sec

DC10 Motorola (C-Port)
EZ Chip NP-2
IBM PowerNP
Intel IXP2800
Vitesse PRISM IQ2200
Cisco Toaster III
Raw Router
VersaBench

www.cag.csail.mit.edu/versabench

Sharing a benchmark set to stress versatility of processors

Categories of programs:
ILP – Desktop and Scientific Streams
Throughput oriented servers
Bit-level embedded
Summary

Raw: single chip for ILP and streaming

Scalar operand network is key to ILP and streams

Tiled processor chip demonstrated in 2002

www.cag.csail.mit.edu/raw