

Fast Benchmarking for Processor Design A Primer

Afzal Hossain, Ph.D.
University of Michigan-Dearborn

Agenda and Opening Remarks

- Motivation
 - Computerized securities markets
 - Smart Grid
 - Digital Media Processor
 - Cloud and Virtualization: *data-center-on-chip*
 - Human Processing Unit (HPU)
- **Key Concepts: Instruction Set Architecture, Micro-architecture, Design Abstraction Layers, Architecture Simulation, Analytical Modeling, CPU Benchmarking**
- **Fast Benchmarking**
- Concluding remarks

A Hossain 2

A Journey from Physics to Social Science

Think beyond technology

A Hossain 3

Computerized Securities Markets Real-time Trades & T+3 Settlement

- Real-time volume transactions
- IP + FIX Protocols
- Fix-point and Floating-point OPs
- Security & encryptions
- Vector, SIMD, VLIW

A Hossain 4

Smart Grid

- Separation of Generation, Transmission, Distribution, and Consumption
- ISOs Manages flow of electrical power
- Manages bulk power market
- **Real-time monitoring, load balancing, and volume transactions**

A Hossain 5

Media Processor- MPEG-2 CODEC Chip

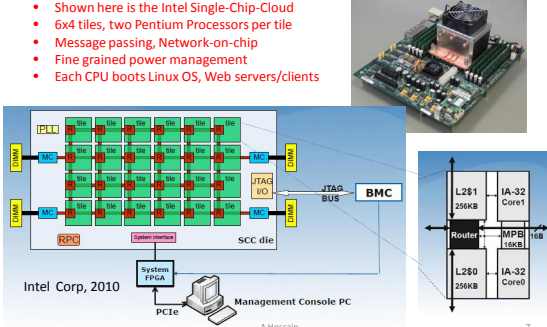
Ishiwata et. al. J. Solid-State Circuits 38(3), 2003

- Heterogeneous multi-processor architecture
- Six customized RISC cores with User Defined Instruction UDI, & HW engines

A Hossain 6


Data-Center-on-A-Chip

- Shown here is the Intel Single-Chip-Cloud
- 6x4 tiles, two Pentium Processors per tile
- Message passing, Network-on-chip
- Fine grained power management
- Each CPU boots Linux OS, Web servers/clients



The diagram shows a 6x4 grid of tiles. Each tile contains two Pentium processors. The system includes a BMC (Baseboard Management Controller) connected via JTAG and System Bus. A System On-Chip (SOC) is also shown with components like PLL, JTAG IO, and BMC. The die is labeled 'SCC die' and 'Intel Corp, 2010'. A Management Console PC is connected via PCIe.

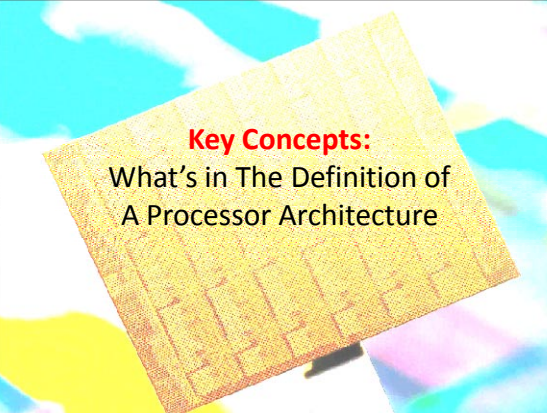
Human Processing Unit (HPU)



The diagram illustrates the HPU concept, showing a person interacting with a computer system. A photo shows a person working at a computer terminal. The text below the photo reads: 'The HPU: James Davis, et al. UC Santa Cruz, June 2010' and 'NASA Photo E49-54'.

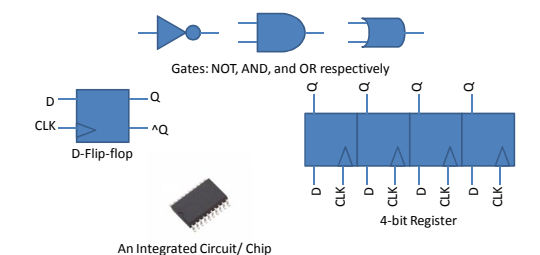
Amazon Mechanical Turk (Dollar for brain)

Key Concepts: What's in The Definition of A Processor Architecture



Digital Electronics Components

Bits: 1, 0 Binary number: b'1100 Hexadecimal number: h'C



Gates: NOT, AND, and OR respectively

D-Flip-flop

4-bit Register

An Integrated Circuit/ Chip

Physical Building Blocks of A Chip

- Digital Logic
 - Combinatorial gates
 - INV, OR, AND, XOR, NAND, NOR
 - Sequential elements
 - Flip-Flops, Latches
 - Combinational logic and finite state machines
- Components
 - Data Path
 - Register file, SRAMs
 - Analog to Digital Converters (ADC)
 - Digital to Analog Converters (DAC)
 - Phase Locked Loops (PLL)

Logical Building Blocks of a System

- Media Applications
 - Satellite, Terrestrial, Cable TV, STB, YouTube, IPTV, VoD, DVD-Player
- Standards & Protocols
 - MPEG-2, H.264, MPEG-4, AC3
 - HTML, XML, CSS-3, Flash, IP, RTP
- Infrastructure Software
 - Operating Systems, DB, C++, Java, Android, Web Browsers
- Processor
 - Media processor, MIPS, Pentium, Sparc, ARM
- Analog & Digital representations
 - 1,0,0,1,1, Voltage, Current, Time, Fields

Levels of Design Abstraction

- Physical design layout, ICs, Boards
- Transistor level design
- Gate level design
- RTL level design
- Micro-architecture specification
- Architecture models
- ISA specification ----- ISA Specs Is The Bridge -----
- OS, DB, Programming Languages
- Application level abstraction

↑ Physical Hardware
↓ Logical Software

A Hossain 13

Example: Basic Computer Instruction Set Architecture (ISA)

Let's time travel to the 1970s for simplicity

- Basic Computer is a 16-bit microprocessor

M. Mano, *Computer System Architecture*, Prentice-Hall 1993

Register	No. of Bits	Register Name	Function
DR	16	Data Register	Holds memory operand
AR	12	Address Register	Holds address for memory
AC	16	Accumulator	Processor register
IR	16	Instruction Register	Holds instruction code
PC	12	Program Counter	Holds address of instruction
TR	16	Temporary Register	Holds temporary data
INPR	8	Input Register	Holds input character
OUTR	8	Output Register	Holds output character

Describe the machine registers in a table

A Hossain 14

Basic Computer Register Diagram

Describe the machine registers including I/O registers using diagrams
Define the memory space using a diagram

A Hossain 15

Basic Computer Instruction Format

15 14 13 12 11 0
 1 | Op code | Address | (Op code: 000 through 110)

a) Memory-reference instructions

15 14 13 12 11 0
 0 | 1 1 1 | Register Operation | (Op code: 111, I = 0)

b) Register-reference instructions

15 14 13 12 11 0
 1 | 1 1 1 | I/O Operation | (Op code: 111, I = 1)

c) Input-Output instructions

Define formats of all machine instructions

A Hossain 16

Memory-reference Instructions

15 14 13 12 11 0
 1 | Op code | Address | (Op code: 000 through 110)

a) Memory-reference instructions

Symbol	I=0	I=1	Description
AND	0XXX	8XXX	AND memory word to AC
ADD	1XXX	9XXX	ADD memory word to AC
LDA	2XXX	AXXX	Load memory word to AC
STA	3XXX	BXXX	Store content of AC in memory
BUN	4XXX	CXXX	Branch unconditionally
BSA	5XXX	DXXX	Branch and save return address
ISZ	6XXX	EXXX	Increment and skip if zero

Using the formats, define each instruction, op-code, and mnemonic

A Hossain 17

Register-reference Instruction

15 14 13 12 11 0
 0 | 1 1 1 | Register Operation | (Op code: 111, I = 0)

b) Register-reference instructions

Symbol	Op Code	Description
CLA	7800	Clear AC
CLE	7400	Clear E
CMA	7200	Complement AC
CME	7100	Complement E
CIR	7080	Circulate right AC and E
CIL	7040	Circulate left AC and E
INC	7020	Increment AC
SPA	7010	Skip next instruction if AC positive
SNA	7008	Skip next instruction if AC negative
SZA	7004	Skip next instruction if AC zero
SZE	7002	Skip next instruction if E is zero
HLT	7001	Halt computer

A Hossain 18

I/O Instruction

c) Input-Output instructions

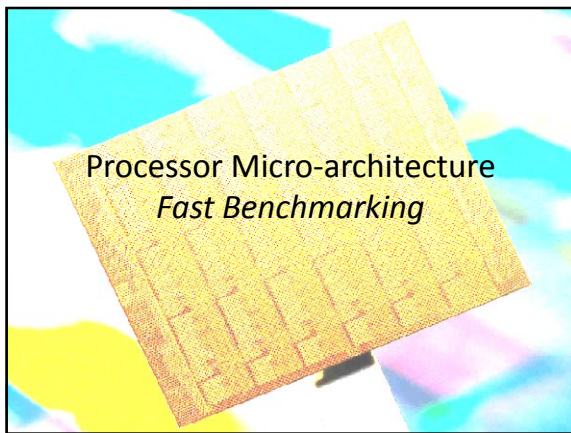
Symbol	Op Code	Description
INP	F800	Input character to AC
OUT	F400	Output character from AC
SKI	F200	Skip on input flag
SKO	F100	Skip on output flag
ION	F080	Interrupt on
IOF	F040	Interrupt off

A.Hossain 19

Components of an Architecture Definition

- **Instruction Set Architecture**
 - Registers, memory space, I/O interface
 - Instructions, op-codes
 - Interrupts, and exceptions handling mechanism
 - Data abstraction and external bus interface
- **Micro-architecture Specification**
 - An ISA can be implemented in many ways. A micro-architecture is an implementation level abstraction of an ISA
 - Complementary to a definition, not a required component

A.Hossain 20



Architecture Simulators

- **What**
 - High level (C, C++, ...) model of processor micro-architecture.
 - Examples: SimPoint, SimFlex, M5, SimpleScalar
 - Choices: Functional, Cycle-accurate, Full-system models, ...
 - Easily available models of - Alpha, ARMS, MIPS, SPARC
 - Branch predict., Caches, LSQ, Out-of-order, speculative exec
- **Why**
 - Micro-architecture performance studies
 - Benchmark performance such as SPEC CPU 2006
 - RTL model validation
- **Benchmarks**
 - SPEC CPU (INT, FP) 2006

A.Hossain 22

Problem

- Performance analysis, and functional validations are the two most important issues of large chip designs
- Architecture simulation can take hours, days
- Data is often hard to interpret
- Performance results are as good as the C/C++ model

A.Hossain 23

Case Study

*A researcher wants to know what size memory is ideal for a unified Trace Cache in a 4-thread SMT processor. He is using the CPU2000 benchmarks with 26 programs. His research design requires 4 independent benchmarks as threads in the SMT simulations. For the given workload, each benchmark has roughly 308 Billion dynamic instructions on average. He is using modified SimPoint with MRRL, where the simulation time is 2.25 hours per thread. His exhibits require a graph showing at least 10 data points clearly establishing the vantage memory size. The researcher wants to consider all possible combination of threads and take averages. In addition, from his prior experience he knows, roughly 50% of all simulations will be wasted if he does nothing to prevent it, so he plans for some prospecting. He wants to estimate simulation runtime. The number of ways 4 threads can be selected from 26 programs, $^{26}C_4 = 14,950$. For each combination he needs 10 simulations for 10 data points. Therefore, he needs 149,500 simulations without wasted runs. He has found a way to reduce wasted runs by selecting 4 arbitrary threads for prospecting the vantage memory size. The prospecting involves 15 exploratory simulations. SMT simulations run slower compared to one thread. Each SMT simulation takes 4 * 2.25 hours or 9 hours to complete. Therefore, the prospecting takes 15x9 or 135 hours of simulation. After prospecting he needs 149,500 * 9 hours for full simulation, that is about 153.6 years, which he considers impractical. So, he decides to simulate only 8 benchmarks, four each from FP and INT. This generates $^{26}C_4$ or 70 simulations and requires 70x9 hours or 26.25 days on a single machine or 2.625 days on 10 distributed machines. Although sampling simulators have known accuracy issues and the benchmarks are partial, the researcher, without other choices, accepts the compromise. ||*

A.Hossain 24

Solution

- Analytical model + Architecture Simulator model
- Analytical mode executes fast (seconds vs. hours)
- Result as good as the model itself
- Good for performance analysis

A.Hossain

25

What's Fast Benchmarking

- A new methodology for micro-architecture performance studies
 - Mathematical models
 - Accurate models predict benchmark performance accurately
 - Fast execution time
 - Minimal use of simulation involved
 - Adaptive model development
 - Adaptive simulations

A.Hossain

26

Model Example

A Trace Cache stores frequently used instructions in program's execution order. Also, a Basic Block is defined as a group of consecutive instructions that starts with the target of a branch instruction, and ends with a branch instruction with no other branch instruction in the middle. The parameter b is the probability that an instruction is a central transferring instruction, therefore the average size of the Basic blocks is $1/b$. A Trace Cache physical line has a fixed length of n instructions.

The first instruction in a Trace Cache line is also the first instruction of the first Basic Block in the line. A Trace Cache line may contain one or more complete Basic Blocks. Accordingly, the last valid instruction in a trace is a branch instruction. The Trace Cache in this example is not allowed to have partial basic blocks, therefore, there are holes or slots of invalid instructions near the end of a line. Fig. 2 shows one such physical line of the Trace Cache. The white slots are valid instructions, the slots with slanted lines are branch instructions, and the dark slots are holes. We want to know what is the average length of valid instructions in a Trace Cache line denoted by the parameter AHL . We will get there by first counting the average number of holes or invalid instructions in the line, denoted by the parameter AHL .

Figure - A Trace Cache physical line

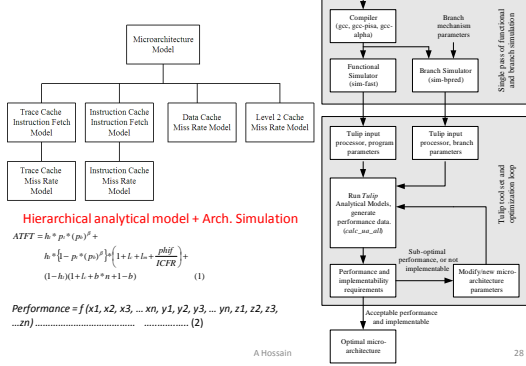
$$AHL = \sum_{i=0}^{n-1} i * (1-b)^i * b$$

$$= \frac{(1-b) - (1-b)^{n+1} * (b * n + 1 - b)}{b} \quad (1)$$

A.Hossain

27

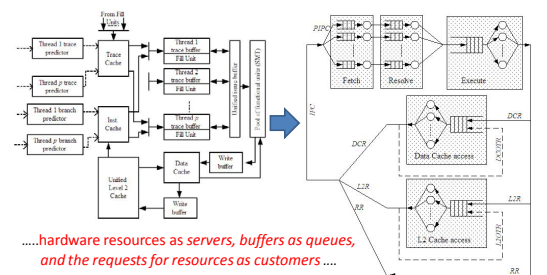
Fast Benchmarking Process



A.Hossain

28

SMT Processor Queuing Model



.....hardware resources as servers, buffers as queues, and the requests for resources as customers

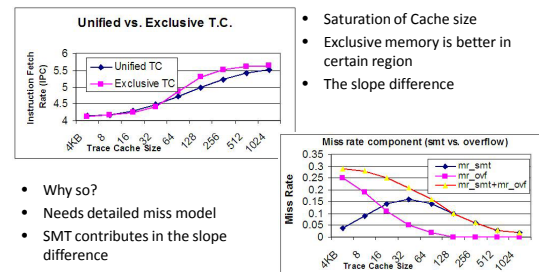
Challenges arise from deviations of randomness ... many queues do not follow a particular discipline... example: instructions can be executed out-of-order. Arrivals of customers, service times of servers do not have simple probability distributions. The arrivals of customers are dependent on each other, because the instructions are dependent on each other. Service time of servers are not randomly distributed for all servers

A.Hossain

29

Study: Unified v. Exclusive Trace Cache

How to organize/allocate memory for Trace Cache in an SMT Processor?



- Why so?
- Needs detailed miss model
- SMT contributes in the slope difference

- Saturation of Cache size
- Exclusive memory is better in certain region
- The slope difference

A.Hossain

30

Comparative Analysis- Accuracy

- TCFR: Trace Cache Instruction Fetch Rate
- Results within ~ 7% of simulation in this case

Benchmark	TCFR Model	TCFR Simulation	Diff.
183.equake	7.2	8.0	-0.8
179.art	6.4	7.0	-0.6
177.mesa	10.4	10.9	-0.5
188.amp	9.0	10.1	-1.1
164.gzip	8.2	7.7	0.5
255.vortex	7.3	8.5	-1.2
181.mcf	7.4	8.0	-0.6
197.parser	3.9	4.1	-0.2

One of the cases with accurate models.
In general, accuracy needs efforts

A Hossain

31

Comparative Analysis- Runtime

- Runtime less than 250 ms vs. hours in this case

Benchmark	Model	Simulation (200 M Inst.)
183.equake	0.25 sec	7040 sec
179.art	0.25 sec	7546 sec
177.mesa	0.25 sec	8338 sec
188.amp	0.25 sec	8426 sec
164.gzip	0.25 sec	7274 sec
255.vortex	0.25 sec	6800 sec
181.mcf	0.25 sec	7440 sec
197.parser	0.25 sec	8894 sec

A Hossain

32

Comparative Analysis- Benefits

- Ease of modeling – pipelining, SMT, Caches, Buses, system, application processors, UDI
- Fast Runtime
- Data interpretation advantages
- Designers often avoid architecture simulations due to long modeling and runtime
 - May imply missed market opportunity window
- Adaptive modeling - Parametric, customizable
- Adaptive simulations

A Hossain

33

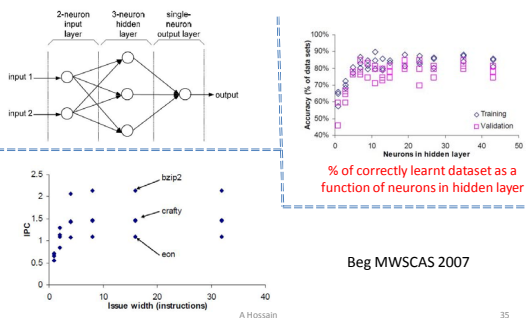
Fast Benchmarking- Research Areas

- Accurate micro-architecture modeling
- Memories, caches, pipelining, speculative/out-of-order-execution
- Heterogeneous systems
- Hardware & Software co-design & partitioning
- Virtualization, Cloud
- Embedded systems

A Hossain

34

Example: Machine Learnt Neural-Network Micro-architecture Model

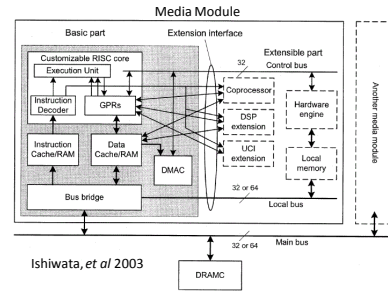


A Hossain

35

Example: HW-SW Partitioning using FB

- SW, HW Engines & UCI, Slow simulations Vs. FB
- $CPU\ Time = Instruction\ count * Clock\ cycle\ per\ instruction * Clock\ cycle\ time$



A Hossain

36

Summary and Q&A

- Applications are the primary drivers of computer research
- Research programs can be developed in the following areas
 - Smart Grid, Cloud Computing, Digital Media Processor, Real-Time high volume securities transactions
- Fast Benchmarking is a new methodology for processor architecture performance analysis