Topic A – Part 3
Dataflow Model of Computation
(From Dataflow to Multithreading)

CPEG 852 - Spring 2014
Advanced Topics in Computing Systems

Guang R. Gao
ACM Fellow and IEEE Fellow
Endowed Distinguished Professor
Electrical & Computer Engineering
University of Delaware
ggao.capsl@gmail.com

3/21/2014
Evolution of Multithreaded Execution and Architecture Models

A version of this slide was presented in my invited talk at Fran Allen’s retirement party July 2002
A Multithreaded Architecture

One PE

To Other PE’s
Case Studies – Dataflow Model Insired Multithreading

- EARTH Model (1993 – mid 2000s)
- The UHPC/Runnemedede Model (2010 - )
McGill Data Flow Architecture Model (MDFA)
Argument –flow Principle

Argument –fetching Principle
A Dataflow Program Tuple

Program Tuple = \{ P-Code . S-Code \}

P-Code

N1: x = a + b;
N2: y = c - d;
N3: z = x * y;

S-Code

\[
\begin{align*}
\text{a} & \quad 2 \quad \text{n1} \\
\text{b} & \quad 3 \\
\text{c} & \quad 2 \quad \text{n2} \\
\text{d} & \quad 3
\end{align*}
\]

IPU ↔ ISU
The McGill Dataflow Architecture Model

- **Pipelined Instruction Processing Unit (PIPU)**
- **Dataflow Instruction Scheduling Unit (DISU)**
  - Enable Memory & Controller
  - Signal Processing
The McGill Dataflow Architecture Model

Important Features

Pipeline can be kept fully utilized provided that the program has sufficient parallelism

Pipelined Instruction Processing Unit (PIPU)

Dataflow Instruction Scheduling Unit (DISU)

Enabled Instructions
Waiting Instructions

Fire
Done

= PC
The Scheduling Memory (Enable)

Dataflow Instruction Scheduling Unit (DISU)

Fire

Count Signal(s)

Enabled Instructions

Waiting Instructions

Signal Processing

Done
Advantages of the McGill Dataflow Architecture Model

• Eliminate unnecessary token copying and transmission overhead.

• Instruction scheduling is separated from the main datapath of the processor (e.g. asynchronous, decoupled).
Von Neumann Threads as Macro Dataflow Nodes

A sequence of instructions is “packed” into a macro-dataflow node.

Synchronization is done at the macro-node level.
The Von Neumann-type Processing

begin
for i = 1 …
…
endfor
end

Source Code

Compiler

Sequential Machine Representation

Load

Processor
Hybrid Evaluation Von Neumann Style Instruction Execution” on the McGill Dataflow Architecture

- Group a “sequence” of dataflow instruction into a “thread” or a macro dataflow node.
- Data-driven synchronization among threads.
- “Von Neumann style sequencing” within a thread.

**Advantage:**
Preserves the parallelism among threads but avoids unnecessary fine-grain synchronization between instructions within a sequential thread.
What Do We Get?

• A hybrid architecture model without sacrificing the advantage of fine-grain parallelism!

(latency-hiding, pipelining support)
A Realization of the Hybrid Evaluation

Pipelined Instruction Processing Unit (PIPU)

Dataflow Instruction Scheduling Unit (DISU)

Shortcut

Fire

Done

Von Neumann bit

1 2 ... k
Case Studies –
Dataflow Model Inspired Multithreading

• McGill Dataflow Model (1988 - 1993)

• EARTH Model (1993 – mid 2000s )

• The UHPC/Runnemede Model (2010 - )
The "hotel" model

Thread Unit
Executor Locus
A Single Thread

CPU
Memory

Coarse-Grain thread-
The family home model

Fine-Grain non-preemptive thread-
The "hotel" model

Coarse-Grain vs. Fine-Grain Multithreading

[Gao: invited talk at Fran Allen's Retirement Workshop, 07/2002]