Algorithm/Architecture Co-design for Smart Signals and Systems in Cognitive Cloud/Edge

Chris Gwo Giun Lee, PhD
Director, Bioinfotronics Research Center, Professor, Department of Electrical Engineering,
National Cheng Kung University
Tainan, Taiwan

Table of Contents

Introduction
- The Big DATA Era
- Analytics Algorithm: analysis of big multimedia & medical data
- Analytic Architecture: analysis of algorithms for SMART SYSTEM design
Algorithm/Architecture Co-design: Analytic Architecture for SMART SoC/Edge
- Architectural Platform before Cloud
- Algorithm/Architecture Co-Design: Abstraction at the System Level
- Dataflow Graph: Modeling the Computational Platform
- Algorithmic Intrinsic Complexity Metrics and Assessment
- Intelligent Parallel/Reconfigurable Computing
Towards IoT, Cloud, & beyond: Big Data Analytics in SMARTECH/AI with Analytic Architecture
- Deep Learning
- Intelligent Surveillance (MPEG)
- Medical Imaging
Introduction

The Math Bridge in Cambridge University
Apollo Navigation Computer

SMARTECH: The New Paradigm Shift in Technology

Industry 1.0: Energy
- Energy
- Industry 1.0: Energy

Industry 2.0: Electricity
- Electricity
- Industry 2.0: Electricity

Industry 3.0: Information
- Information
- Industry 3.0: Information

Industry 4.0: AI
- AI
- Industry 4.0: AI

Fast Changing Landscape: Engineers have been the center of ALL these technologies from the very beginning!
Data Gets Ever Larger

Marshall McLuhan (1960’s):

*Electronic Media, primarily Television being extension of human nervous systems*

Today:

*Extending outwards even further with multiple sensors interconnected. When going deeper inwards into the human body with huge data from human brain and human genome*

---

Reaching Out Even Further via IoT & Going in Ever Deeper into the Human Brain and Genome
Ever More Complex Analytics Algorithms Should Run on Analytics Architecture

Analytics Algorithm: Analyzes speech & images

Speech recognition with feelings

Facial emotion detection

Analytics Architecture: Analyzes algorithms & data

Algorithm/Architecture Co-Design: Analytics Architecture for SMART SoC
New Design Paradigm: Moving from programming to design and beyond... post Moore's law

Wirth from ETHZ (1975):

Programming = Algorithm + Data Structure

Lee from NCKU (2007):

Design = Algorithm + Architecture

Architectural Platforms Beyond Cloud

Towards Cloud & Heterogeneous Systems
- Embedded general purpose instruction set processor
- Embedded multicore processor/GPU
- Instruction Set DSP
- Application Specific Instruction Set Processor (ASIP)
- Reconfigurable Processor/FPGA
- Embedded Reconfigurable Logic/FPGA

Performance/Area

Flexibility

Power
Cross Level of Abstraction Design Space Exploration

Application/Specification
Algorithm
Architecture
Cycle-accurate model
Synthesized netlist
Physical design
Device level
Different instances or realizations

Application
System
Architecture
Microarchitecture
Circuit
Device

Algorithm/Architecture Co-exploration

Algorithm/Architecture Co-Design: Abstraction at the System Level

Algorithm Design

Complexity Measurement
No. of operations
Degree of parallelism
Data transfer rate
Data storage requirements

Data Flow

Back Annotation

Architecture Design

Futuristic Smart System Design Methodology

Algorithm Space Exploration

Algorithm/Architecture Co-exploration (Specific Graph Theory)

Software/Hardware Co-exploration

Circuit/Device Co-exploration

Modelling the Computational Platform via Dataflow Graphs (DFG)

Applications:
- IoT
- Mobile
- Communication
- Multimedia
- Biomedical

Algorithm/Space Exploration
- Computing algorithms
- Algorithmic complexity
- Characterisation of algorithmic complexity

Algorithm/Architecture Co-exploration
- Dataflow modeling at different abstraction levels
- Design flow
- Architectural information
- Parallel/Reconfigurable

Software/Hardware Co-exploration
- Programmable fabric (CPU/GPU design)
- Reconfigurable Logic design
- Customised logic

Circuit/Device Co-exploration
- MOS transistor design
- Memory device design
- Analog VLSI

Back annotation
- Delay model, energy consumption, circuit reliability

Device capacitances, Memory retention, endurance & read/write latency & current

MODELING the COMPUTATIONAL PLATFORM via Dataflow Graphs (DFG)
Three-tap FIR filter: \( y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] \)

**Direct form:**

- \( x[n] \) inputs
- \( a_0, a_1, a_2 \) coefficients
- \( z^{-1} \) delays
- \( y[n] \) output

**Transposed form (Data broadcast form):**

- \( y[n] \) inputs
- \( a_0, a_1, a_2 \) coefficients
- \( z^{-1} \) delays
- \( x[n] \) output
Pipeline View of Dataflow in FIR (Direct Form)

Clock Cycles | Input | $D_0$ | $D_1$ | Output
---|---|---|---|---
0 | $x[0]$ | | | $y[0] = a_0x[0]$ |
1 | $x[1]$ | $x[0]$ | | $y[1] = a_0x[1] + a_1x[0]$ |
... | | | | |
127 | $x[127]$ | $x[126]$ | $x[125]$ | $y[127] = a_0x[127] + a_1x[126] + a_2x[125]$ |
128 | $x[127]$ | $x[126]$ | $x[125]$ | $y[128] = a_0x[127] + a_1x[126]$ |
129 | $x[127]$ | $x[126]$ | | $y[129] = a_0x[127]$ |

Pipeline View of Dataflow in FIR (Transposed Form)

Clock Cycles | Input | $D_0$ | $D_1$ | Output
---|---|---|---|---
0 | $x[0]$ | | | $y[0] = a_0x[0]$ |
1 | $a_0x[0]$ | $x[0]$ | | $y[1] = a_0x[1] + a_1x[0]$ |
2 | $a_1x[1]$ | $a_0x[1]$ | $x[0]$ | $y[2] = a_0x[2] + a_1x[1] + a_2x[0]$ |
... | | | | |
127 | $a_0x[127]$ | $a_0x[126]$ | $a_1x[126]$ | $y[127] = a_0x[127] + a_1x[126] + a_2x[125]$ |
128 | $a_0x[127]$ | $a_0x[126]$ | $a_1x[126]$ | $y[128] = a_0x[127] + a_1x[126]$ |
129 | $a_0x[127]$ | | | $y[129] = a_0x[127]$ |
Synthesis Results for 3Tap FIR Filters Using Direct and Transpose Forms

<table>
<thead>
<tr>
<th>architecture criteria</th>
<th>Direct Form</th>
<th>Transposed Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area (gate-count)</td>
<td>1212 (1110, 101)</td>
<td>1674 (1573, 101)</td>
</tr>
<tr>
<td>Timing (ns)</td>
<td>12.7 ns</td>
<td>10.37 ns</td>
</tr>
<tr>
<td>Power (mW)</td>
<td>35.03 mW</td>
<td>37.26 mW</td>
</tr>
</tbody>
</table>

(1) Gate count is in terms of equivalent 2-input NAND gate
(2) Timing is based on static timing analysis
(3) Power dissipation is only a very rough estimation

Dataflow Graph Models

- They should contain:
  - Algorithmic information or behavior
  - Architectural information for implementation
- Some important dataflow models are:
  - Directed acyclic graph (DAG)
  - Synchronous dataflow (SDF) graph
  - Control data flow graph (CDFG)
  - Kahn process networks (KPN)
  - Y-chart application programming interface (YAPI)
## Fundamental Dataflow Models

<table>
<thead>
<tr>
<th>Methods</th>
<th>Descriptions</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directed acyclic graph (DAG)</td>
<td>Vertex: atomic operations</td>
<td>Dependency</td>
</tr>
<tr>
<td></td>
<td>Directed edge: data dependency</td>
<td></td>
</tr>
<tr>
<td>Synchronous dataflow (SDF) graph</td>
<td>Extension of DAG by annotating edges with static data rates</td>
<td>Data rate</td>
</tr>
<tr>
<td>Control data flow graph (CDFG)</td>
<td>Encapsulating DAGs as vertexes controlled by control edges</td>
<td>Control flow</td>
</tr>
<tr>
<td>Kahn process networks (KPN)</td>
<td>Actor: atomic operations or processes</td>
<td>Data-driven</td>
</tr>
<tr>
<td></td>
<td>Communication among actors: first-in-first-out (FIFO)</td>
<td>Concurrence</td>
</tr>
<tr>
<td>Y-chart application programming interface (YAPI)</td>
<td>Extension of KPN to support dynamic selection of FIFO channels</td>
<td>Controllability</td>
</tr>
</tbody>
</table>


### Directed Acyclic Graph (DAG)

- DAG are acyclic graphs composed of nodes and directed edges:
  - nodes indicate operation activities
  - directed edges between nodes represent the dependency of operations.
- Arrows indicate the directions in which the data flows through the operation nodes

```
A -- B -- C -- F
     |     |
     D -- E
```

A, B…F: operations of the nodes
**Synchronous Dataflow (SDF) Graph**

- **SDF** are graphs extended from DAGs where specified a priori
  - *Heads* of edges represent fixed input
  - *Tails* represent output data-rates of nodes,
  - If the input/output data rate of the nodes are not specified or known a priori, we then have asynchronous dataflow graph

![Diagram of Synchronous Dataflow Graph](image)

- A, B…F: operations of the nodes
- a, b,…o, p: data-rate of the nodes.

**Control Dataflow Graph (CDFG)**

- CDFG describe the flow of data computation and control procedures using **nodes, control edges** and **data edges**.

Pseudo codes:

```plaintext
1  DFG_0
2  if (condition_1)
3    DFG_1
4  else if (condition_2)
5    DFG_2
6  DFG_3
```

![Diagram of Control Dataflow Graph](image)
Kahn Process Networks (KPN)

- KPN are directed graphs consisting of nodes and edges. Nodes indicate processes and directed edges represent unbounded and infinite FIFO queues between processes:
  - Blocking read: a process is fired when input data exist.
  - Non-blocking write: output data are sent out via a unbounded FIFO.

---

Y-chart Application Programming Interface (YAPI)

- Extension of KPN. KPN models deterministic (synchronous) dataflow.
- YAPI is capable of modeling non-deterministic (asynchronous) data flow by allowing dynamic decision on selecting which channel to communicate.
- The selection of port/channel depends on the amount of data.

Mechanism:
Let \( c(p_k) \) be the number of tokens existing in port \( p_k \) and \( N_k \) be the number of tokens required to be consumed from port \( p_k \) at a time.

\[
\text{if } ( c(p_k) \geq N_k) \text{ then read } N_k \text{ tokens from port } p_k
\]
How Big is Big?

Algorithmic Intrinsic Complexity
Metric & Assessment:
PLATFORM INDEPENDENCE

Number of Operations

- Estimates the number of each type of operations
  - Addition/Subtraction
  - Multiplication
  - Division
  - Shift
  - Logic operations
- Operations with constant input and variable input should be differentiated to provide high accuracy
  - $X + Y$ vs. $X + 5$ (X and Y are variables)
  - $X \boxplus Y$ vs. $X \boxplus 5$ (X and Y are variables)
- In addition, the precision of each operand should be taken into account, since it can significantly influence complexity
SMART TRANSFORM PAIR via Spectral Graph Theory for Intelligent Parallel/Reconfigurable Computing

Parallel Computing (Forward Transform): Efficient & Flexible Cognitive Cloud

In GlobalSIPS 2015, IBM T.J. Watson, Intel, etc. consistently addressed this technology as THE future of cognitive computing


Reconfigurable Computing (Inverse Transform): Efficient & Flexible Mobile Edge

Technology adopted by ISO/IEC/MPEG’s Reconfigurable Ad Hoc Group & served as Project Lead


Spectral Graph Theory (SGT):

Graph

Adjacent matrix \( A \)

\[
A(i,j) = \begin{cases} 
1 & \text{if vertex i and vertex j are adjacent to each other} \\
0 & \text{otherwise} 
\end{cases}
\]

\[
A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}
\]

Laplacian matrix \( L = D - A \), where \( D \) is a diagonal matrix where the diagonal elements represents the number of edges connected to that node.

\[
L(i,j) = \begin{cases} 
D(i,j) & \text{if } i = j \\
-1 & \text{if } i \neq j \text{ and vertex } i \text{ is adjacent to vertex } j \\
0 & \text{otherwise} 
\end{cases}
\]

\[
L = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}
\]
**Degree of Parallelism: Eigen-Analysis of Dataflow Graphs using SGT**

**Algorithm**

\[ O_1 = A_1 + B_1 + C_1 + D_1 \]
\[ O_2 = A_2 + B_2 + C_2 + D_2 \]

**Dataflow diagram**

\[ \begin{align*}
A_1 & \rightarrow \bullet \\
B_1 & \rightarrow \bullet \\
C_1 & \rightarrow \bullet \\
D_1 & \rightarrow \bullet \\
& \rightarrow \bullet \rightarrow O_1 \\
A_2 & \rightarrow \bullet \\
B_2 & \rightarrow \bullet \\
C_2 & \rightarrow \bullet \\
D_2 & \rightarrow \bullet \\
& \rightarrow \bullet \rightarrow O_2
\end{align*} \]

**Causation graph**

\[ \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & -2 & 0 & 0 \\
-1 & -1 & 0 & 0 & 0 & 0 & 0 & 2 \\
\end{pmatrix} \]

**Laplacian matrix**

\[ \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & -2 & 0 & 0 \\
-1 & -1 & 0 & 0 & 0 & 0 & 0 & 2 \\
\end{pmatrix} \]

**Parallelism**

\( 2 \) (Homogeneous)

Quantification of parallelization, Instruction Set Architecture (ISA) design

---

**Reconfigurable Architecture: Commonality Extraction from Dataflow Graphs**

- Observe the common parts between each dataflow graph.

**MPEG-4 filter coefficients**

\([-8,24,-48,160,160,-48,24,-8]\)
\([-1,3,-6,20,20,-6,3,-1]\)

- Divide by 8

**AVC/H.264 filter coefficients**

\([1,-5,20,20,-5,1]\)

**MPEG-4 Chroma interpolation, MPEG2, and AVC/H.264 chroma prediction**

\[ \begin{pmatrix}
1 & 1 \\
1 & 1 & 1 \\
\end{pmatrix} \]
Reconfigurable fractional interpolation

Data Storage: Configuration & Requirement using Graph 1/2

- Each initial cell region confined by external nuclei markers is represented by a vertex.
- Images are divided into tiles of size 64 by 64.
- The vertex size is defined as the number of tiles that each region is contained in.
  - The tile size is obtained from the statistical average of all initial cell region areas.
- Fixed tile size is introduced for efficient parallel processing of irregular cell regions.
- The tile size could also be considered together with the platform architecture for computational efficiency.
Data Storage: Configuration & Requirement using Graph 2/2

- The weight on each edge represents the proportionate overlapping area between the two cell regions represented by the corresponding vertices.
- The weight of each edge is defined as:
  \[ w_{i,j} = \frac{\text{Number of Overlapped Tiles}_{i,j}}{\text{Vertex Size}_i + \text{Vertex Size}_j} \]
- This edge weight indicates the tendency of grouping or compiling corresponding tiles of the two vertices into the same computation cluster so as to optimize data storage, maximize data reuse and minimize data transfer rate.
- The tile size could also be considered together with the platform architecture for computational efficiency.

Data Transfer Rate & Bandwidth via Graph Theory

\[
y[n] = x[n-1] + x[n]
\]

\[
x[n-1] \quad x[n]
\]

\[
y[n]
\]

\[
\begin{align*}
v_1 & = 1 \\
v_2 & = 0 \\
v_3 & = -1 \\
v_4 & = 0
\end{align*}
\]

\[
\begin{bmatrix}
e_1 \\
e_2 \\
e_3
\end{bmatrix}
\]

\[
\begin{bmatrix}
v_1 & v_2 & v_3 & v_4 \\
1 & 0 & -1 & 0 \\
0 & 1 & -1 & 0 \\
0 & 0 & 1 & -1
\end{bmatrix}
\]

\[
Mx = \begin{bmatrix} e_1 \\
e_2 \\
e_3 \\
e_4 \end{bmatrix}
\]

1/30/2019 Bioinfotronics Research Center  Prof. Gwo Giun Lee/NCKUEE  M SoC Lab. 39
A Very Useful SMART Sensor System

SMARTLET: SMART toiLET

SMART is a BUZZ word that SELLS

IoT: Any systems connected via signals for the exchange of information
System Platforms: From System-on-Chip to IoT/Cloud & Beyond

- SoC Platforms
- IoT and Networking Platforms
- Intelligent Surveillance (IS)
- Intelligent Health Cloud (IHC)

Distributed System

- The central control system performs pending jobs analysis:
  - Similarity and loading of pending jobs
  - Overhead of communication
  - Workload of processors
- **Intelligently allocate jobs to processors/clients** (located at different physical locations) based on the analyzed parameters
- When reconfiguring the internet architecture or platform, we have cognitive internet
Cloud Computing/Service

- According to the priority and difference of requests from clients
- Cloud server *intelligently reschedules and reallocates the resource in response to the clients.*

![Cloud Computing/Service Diagram](image)

Storage in Cloud Server

- Analyze the similarity of requests and evaluate the workload of processors
- *Intelligently reallocate and reschedule the storage resource to the available processors*

![Storage in Cloud Server Diagram](image)
Reconfigurable Computing over Cloud
Similarity/Correlation Analysis: Telemedicine

Graph → Similarity/Correlation analysis to reconfigure scalable computing clusters

Deep Learning
Modeling the Neuron Mathematically

- Neural network could define a complex, non-linear form to represent the relationship between input $x$ and output $y$ through a hypothesis $h_{w,b}(x)$
  - Neuron is the computational unit that represents input data in another way
  - $h_{w,b}(x) = f(w^T x + b)$, $f: \mathbb{R} \rightarrow \mathbb{R}$ is activation function.
  - Popular activation functions, sigmoid, hyper tangent, rectified linear function

![Neuron Diagram]

- $x = [x_1, x_2, \ldots, x_N]^T$  
  - $w = [w_1, w_2, \ldots, w_N]^T$

Its All about Synapses of Neurons in Forming a Network

- Multi-Layer Perception (MLP)
  - This example includes one input layer, one hidden layer and one output layer.
  - Forward propagation to predict output based on input.
  - Back propagation to tune neural network based on the prediction error.
  - It can be configured in several hidden layers, different connectivity between layers.
Convolutional Neural Network (CNN)

- Instead of using *fully-connected* neural network, using *locally-connected* neural network to extract local features.
  - Includes convolution layer, pooling layer, fully-connected network
- Training method:
  - Unsupervised pre-training (stacked (denoising) autoencoder)
  - Back propagation through label information

Deep Belief Nets (DBN)

- Deep Belief Nets (DBN):
  - Restricted Boltzmann Machine (RBM) + directed belief nets
- Visible layer: input data
- Hidden layer: features
- Training method:
  - Unsupervised pre-training (stacked RBMs)
  - Back propagation for fine-tuning
- DBN could also be multi-dimensional structure.
Computer Scientist's Perspective of the Human Brain

Collaboration with IBM TJ Watson Research Center

![Brain Diagram](image)

International Standardization in ISO/ITU/MPEG

- Activities with Academia/Industry Worldwide: MIT, Aachen, EPFL, MERL, Qualcomm, Samsung, LG, Sharp, MediaTek, etc

![Standardization Diagram](image)

Encoded bitstream → Syntax → Reconfigurable Video Coding

3D-HEVC

Decoded video

Reconfigurable Video Coding
Biomedical Imaging: Harmonically Generated Microscopy

Analysis of Harmonic Generation Microscopy Image from NTU, Taiwan including CNN

Cross section of human skin

Epidermis

Stratum corneum (SC)
Stratum granulosum (SG)
Stratum spinosum (SS)
Stratum basale (SB)

Dermis

Seborrheic keratosis (SK) & STEM cell detection
• Observe through the image stack in the epidermis, because of the proliferation of basaloid cell, result in acanthotic epidermis.
• In normal skin, epidermal thickness: 60–80um.
• In SK, epidermal thickness: exceed 100um.

BCC Detection via Deep CNN

• Detection of Basal Cell Carcinoma (BCC) via Deep CNN using ~2000 HGM Images with 97.02% accuracy, 95.08% specificity, and 99.04% sensitivity.

• Transfer Medical Doctor’s knowledge to feature layers of Deep Convolution Neural Network (CNN)
SMART Cloud Computing for Precision Medicine

- Patents licensed by Inform Genomics in USA on Analytics Architectural Platform

Chemotherapy Induced Nausea Vomiting

- Data Analytics for CIPN NCKU Medical School and Hospital
Sir would like to edit descriptions.

Chen, 3/27/2016
Intelligent Health Cloud Database Management System

- Collaboration with Banner Alzheimer’s Institute, AZ, USA targeted for Telemedicine

Conclusion

We need to have cross level of abstraction system design methodology to solve cross disciplinary problems in SMART manners

- CLOUD/EDGE/Neuromorphic Computing
- Intelligent Surveillance for Autonomous Cars
- Precision Medicine
- Brain Initiatives……
- Nanoscale Histopathology
Computer, Communication, Control & Care