Introduction of Processor Design for AI Applications

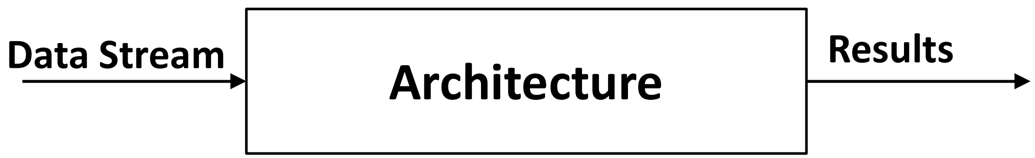
L03 – Graphical Representations

Pengju Ren
Institute of Artificial Intelligence and Robotics
Xi'an Jiaotong University

http://gr.xjtu.edu.cn/web/pengjuren

Typical Data-Stream Algorithms

Algorithms	System Applications
Speech Coding/Decoding	Personal Communication, Multimedia
Speech Encryption/Decryption	Personal Communication, Secure Communications
Speech Recognition	Computer-Human Interface, Robotics
Speech Synthesis	Multimedia, Robotics
Video/Image Detection	Computer Vision, Multimedia, Robotics
Noise cancellation	Professional audio, TWS
Image Compression/Decompression	Digital Cameras, Multimedia
Beamforming	Navigation, Radar/Sonar, Signals Intelligence
Echo Cancellation	Speakerphones, telephone Switches



Linear time-invariant systems

Typical Data-Stream Algorithms

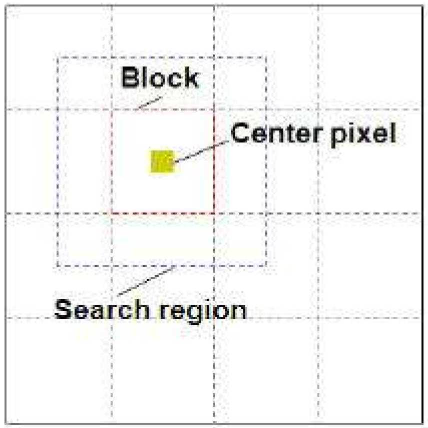
Convolution: $$y(n) = x(n) * h(n) = \sum_{k=0}^{\infty} x(k)h(n-k)$$

Correlation: $$y(n) = \sum_{k=0}^{\infty} h(k)x(n+k)$$

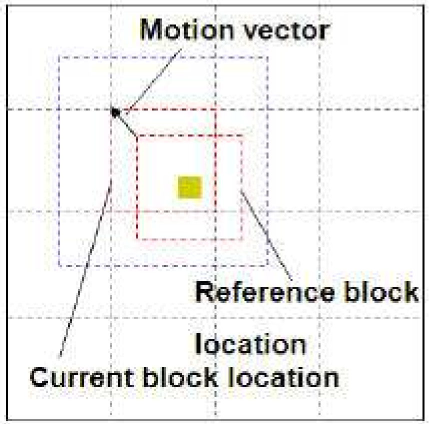
Digital Filters: $$y(n) = \sum_{k=0}^{M-1} b_k x(n-k)$$

Motion Estimation: $$s(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |x(i,j) - y(i+m, j+n)| \quad for \ -p \le m,n \le p$$

(Mean Absolute Difference, MAD)



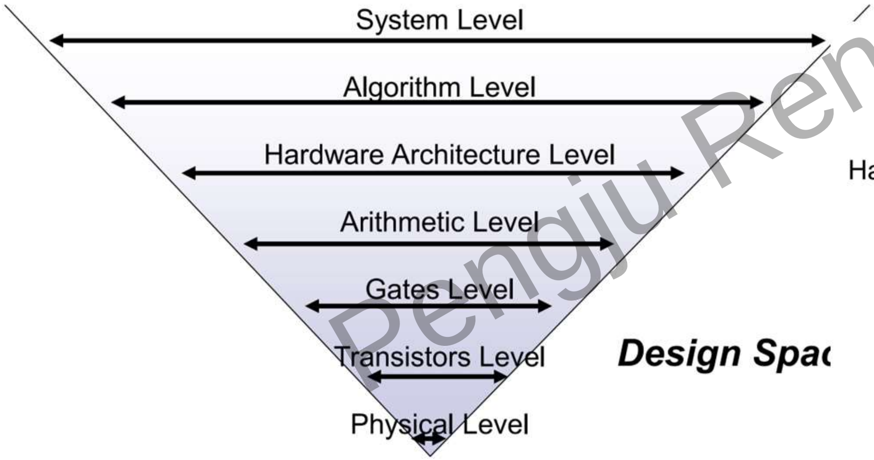
Current frame n

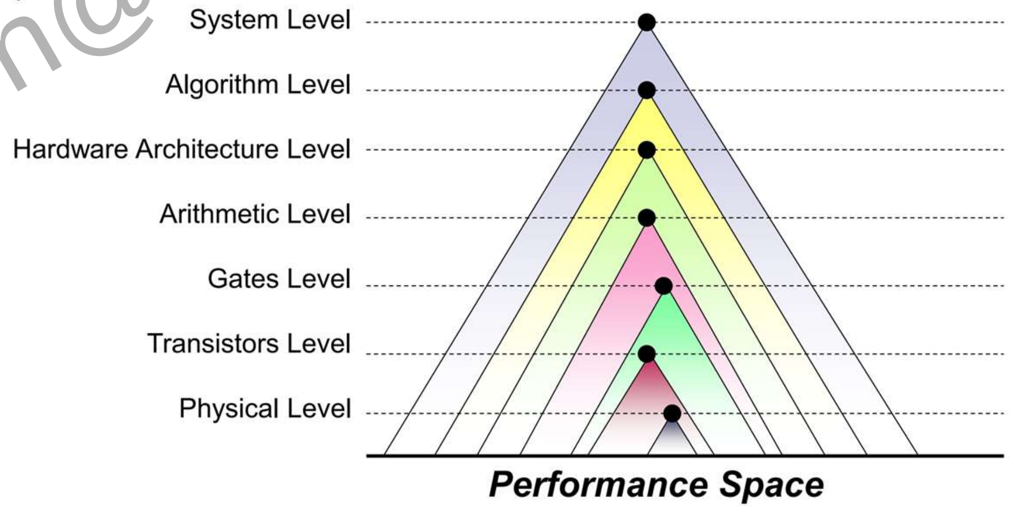


Reference frame n-1

Data-Stream Architecture Design

- Given Data-Stream algorithms, find the "suitable" solution in the design space under certain constraints
- Alternatively, modified or develop the algorithm to be "hardware oriented"/"hardware friendly", and then develop the hardware architecture





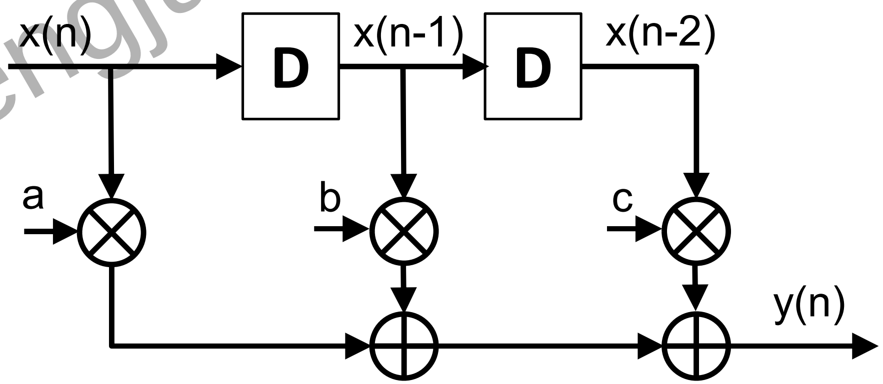
The Higher the abstraction, the larger design space and the more important

Therefore, we need a property representations for Data-stream algorithms

Graphical Representations: Block Diagram (BD)

$$FIR: \quad y(n) = a * x(n) + b * x(n-1) + c * x(n-2)$$

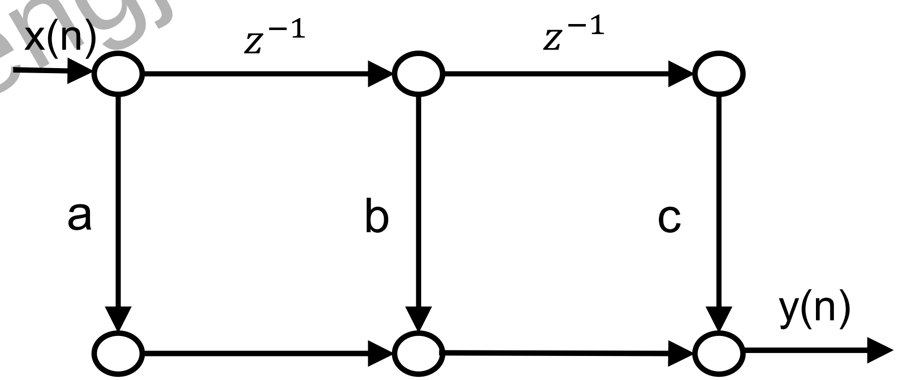
Block Diagrams(BD): Consists of functional blocks connected with directed edges, which represent data flow from its input block to its output block . **BD can be constructed for all systems with different levels of abstraction**.



Graphical Representations: Signal-Flow Graph (SFG)

$$FIR: \quad y(n) = a * x(n) + b * x(n-1) + c * x(n-2)$$

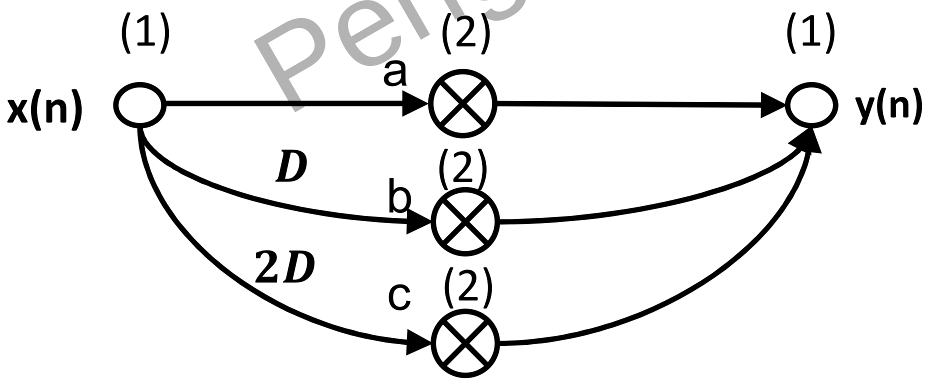
- Nodes: represent computations and/or task, sum all incoming signals
- Directed edge (j, k): denotes a linear transformation from the input signal at node j to the output signal at node k
- Linear SFGs can be transformed into different forms without changing the system functions. For example, ***Flow graph reversal*** or ***transposition*** is one of these transformations (Note: only applicable to single-input-single-output systems)
- Usually used for *linear time-invariant* DSP systems representation

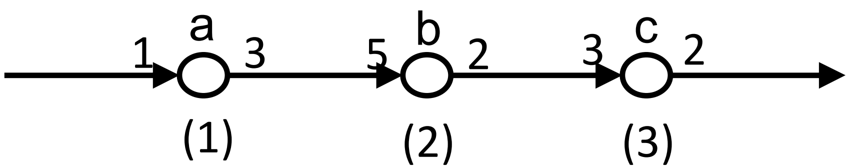


Graphical Representations: Data-Flow Graph (DFG)

$$FIR: \quad y(n) = a * x(n) + b * x(n-1) + c * x(n-2)$$

- DFG: nodes represent computations (or functions or subtasks), while the directed edges represent data paths (data communications between nodes), each edge has a nonnegative number of delays associated with it.
- DFG captures the **data-driven property** of DSP algorithm: any node can perform its computation whenever all its input data are available.
- Each edge describes a precedence constraint between two nodes in DFG:
  - **Intra-iteration precedence constraint:** if the edge has zero delays
  - **Inter-iteration precedence constraint:** if the edge has one or more delays
  - DFGs and Block Diagrams can be used to describe both linear single-rate and nonlinear **multi-rate** DSP systems





Two More Things

- How to determine **the size of FIFO** between two nodes to avoid overflow and data loss?
- How to **matching nodes** with different number of samples produced or consumed ?

Next Lecture： Iteration Bound

Recap：Data streaming Algorithms

Algorithm：“a step-by-step procedure for solving a problem or accomplishing some end”

- Data streaming Algorithms (e.g. DFT, DCT, DNN, Compression, Encryption, NLP)
  - Numerically intensive
    - **Number representation**
    - Bit-width
  - Limited or no control-flow (branching)
  - Streaming/Samples:
    - From ADC, File storage, Camera …
    - Rate often determined by physical factors

Number Representation

Integer: 0110 1010b = 106

Real value: 0110.1010b = 6.625 = $\frac{106}{2^4}$

Integer *Fraction*

Dynamic range

8-bit Integer: smallest +value: 1, largest +value: 127

Dynamic range = 127/1 = $2^7 - 1$

8-bit 4.4 numbers: smallest +value : 2^{-4} , largest +value: 7.9…= $2^3 - 2^{-4}$

Dynamic range = $(2^3 - 2^{-4})/ 2^{-4} = 2^7 - 1$

e.g 32-bit fixed float number: Dynamic Range = $2^{31} - 1$

Scientific notation and floating point

Scientific notation

3.14 x 10^{0}

6.18 x 10^{-1}

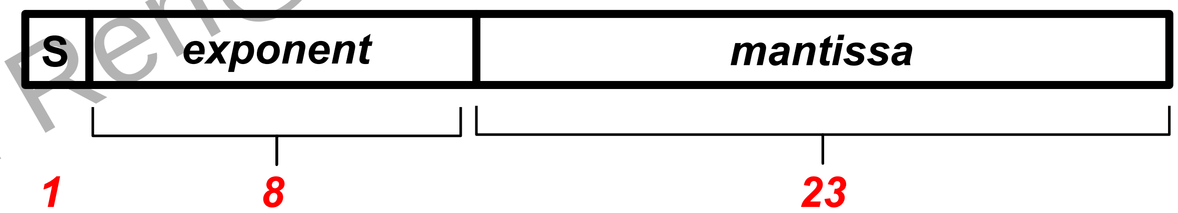
1.52 x 10^{15}

3 digits ***Exponent -> Scaling factor***

[1.mmm…m] x 2^{e-12}

Min = 1.0…0 = 1

Max = 1.111..1 = $2 - 2^{-2} \approx 2$



Dynamic range (DR):

smallest +value: S=0, e = 1(0 reserved) => $2^{1-127} = 2^{-126}$

m =0 => Value = 1.0 x $2^{-126} \approx 10^{-40}$

Largest value: S=0, e = 254 (255 reserved) => $2^{254-127} = 2^{12}$

m =111..1 => Value = 1.999… x $2^{12} \approx 10^{40}$

DR = 10^{80} Single Precision floating point, IEEE 754 Standard