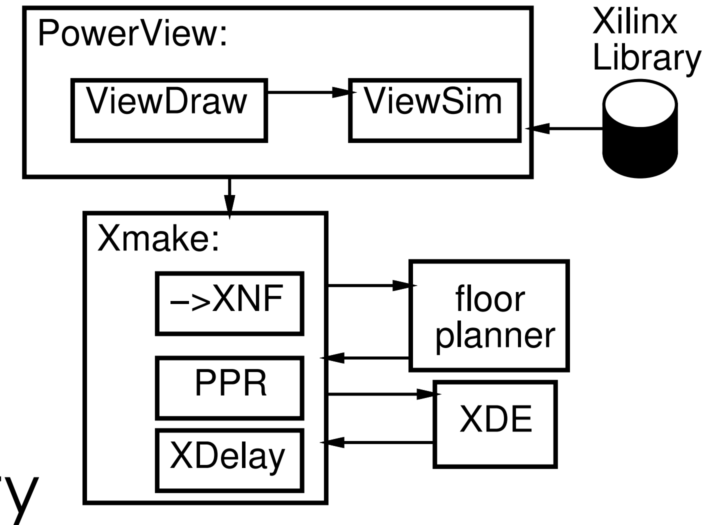
CS294-7: Reconfigurable Computing

Day 8: Thursday, February 13

Spatial Computing Styles

- Admin
- Introduction to Systolic Computing

Admin



- **Project 1 Proposals due**
- **Comment on readings**
- **CAD Tools for Project 1**
  - **Today**: Handouts on schematic entry and library
  - **Accounts**:
    * go to any unix instructional lab (Cory 105, 199; SODA 271, 273, 275, 306)
    * instructions on BBoards for account creation (should get same username as for dept. account)
    * will be able to access machines remotely (see handout)

Systolic

sys.to.le *n*: a rhythmically recurrent contraction; *esp*: the contraction of the heart by which the blood is forced onward and the circulation kept up
— **sys.tol.ic** *adj*

– Merriam Webster (10th)

Systolic architectures are characterized by regular (rhythmic) dataflow through numerous processing elements.

Motivation

Accelerate compute-bound problems

For each data item read from memory,
use it multiple times.
(compute on intermediates without
reading/writing to memory)

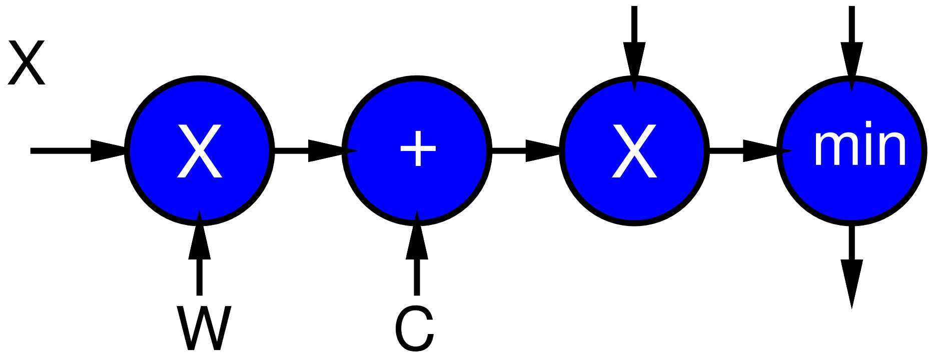
Philosophy

Match VLSI costs:

- Exploit Concurrency
- Wires take up space
- Traversing long wires takes significant time
- Exploit regularity
  - minimize design effort
  - scale w/ performance requirements (problem size)
- (Instructions take up space)

Assembly Line View

Can think of it as an assembly line of processing tasks operating on data as it flows by.



Systolic Architectures

Goal: General methodology for mapping computations into hardware (spatial computing) structures.

Composition:

- **simple compute cells (*e.g.* add, subtract, max, min, ...)**
- **interconnected in regular pattern**
  - **array (linear, 2D)**
  - **tree**
- **pipelined communication between cells**
- **I/O at boundaries**

Why Study?

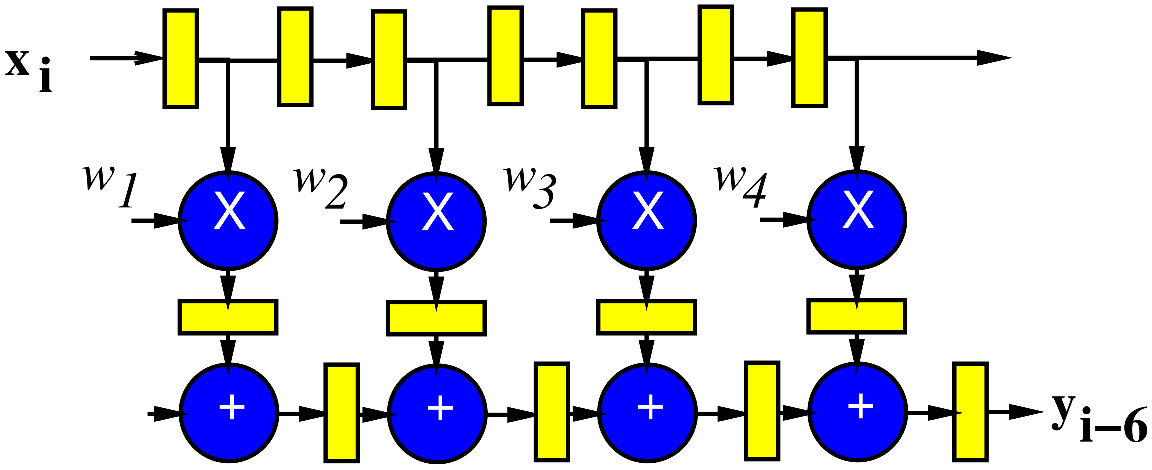
Original theory was to build specialized accelerators — custom hardware for each problem class.

- **Model/goals is a close match to costs in our reconfigurable devices**
- **Many good solutions developed (should know in our bag-of-tricks)**
- **Transformational techniques valuable to RC**

Today: Introduction
Tuesday: More Examples

Finite Impulse Response Computation

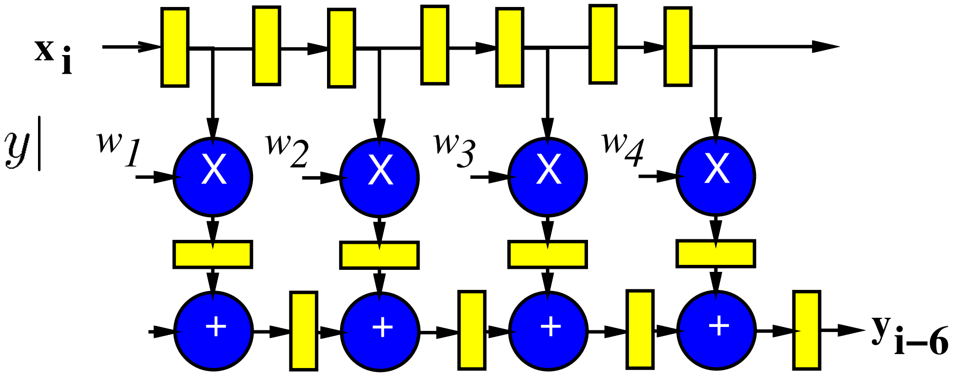
$$y_i = w_1 \cdot x_i + w_2 \cdot x_{i+1} + \cdots + w_k \cdot x_{i+k-1}$$



Finite Impulse Response Computation

Sequential

- **Input BW:** $|w| + |x|$ / TAP ($k \cdot (|w| + |x|)$ / FIR)
  (reads each x k times; reads each w once per result)
- **Output BW:** $|y|$ / FIR
- **Storage is** $k \cdot (|w| + |x|) + |y|$



Systolic

- **Input BW:** $|x|$ / FIR
- **Output BW:** $|y|$ / FIR
- **Each input value used** k **times (**k **multiply-adds)**
- **Internal storage is** $k \cdot \left(|w| + |y| + 2 \cdot |x|\right)$
  ($\approx 2.5\times$; not $k\times$)

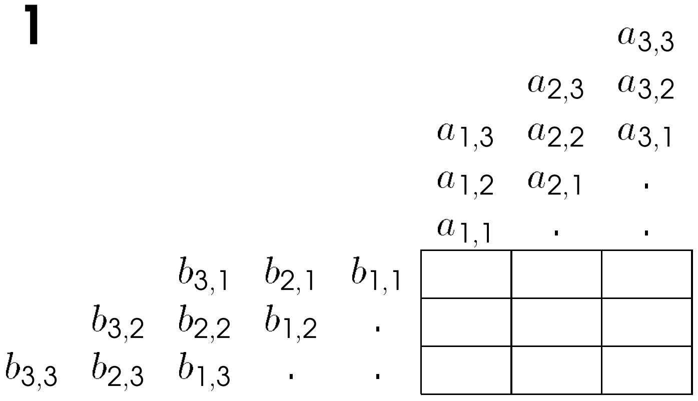
Matrix Multiply

$n \times n$ **array:**

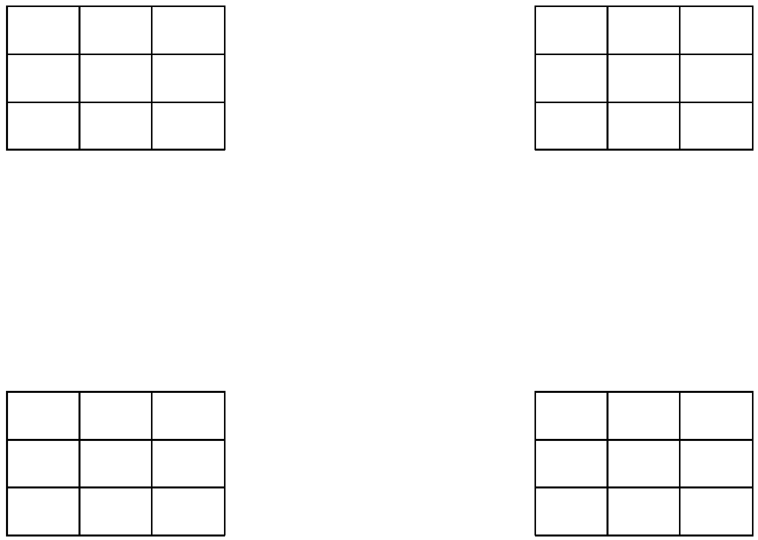
$$C = A \cdot B$$

$$c_{i,j} = \sum_{k=1}^{N} a_{i,k} \cdot b_{k,j}$$

```
if (reset)
    sum = Xin*Yin;
else
    sum = sum+Xin*Yin;
Yout=Yin;
Xout=Xin;
;; omit exit details
```



Matrix Multiply



Systolic Matrix Multiply

matrix multiply	$n \times n$
cells	$n \times n$
latency (including output)	$4n$
MACs	n^3 in n ($4n$) cycles
i/o BW	$2n$/cycle
storage	3 values/cell $(3n^2\|w\|)$

With data in matrix
can pipeline to compute a new result every n cycles

Application

- Signal and image processing
  - **FIR**, IIR, 1- and 2-D convolution, correlation
  - DFT, DCT
  - interpolation
  - median filtering
  - geometric warping
- Matrix arithmetic
  - matrix-vector, **matrix-matrix multiplication**
  - triangularization
  - QR decomposition
- Non-numeric
  - stack, queue, search, sort, **priority queue**
  - graph algorithms
  - language recognition
  - **dynamic programming**
  - **relational database**
  - encoders

Limitations

- each compute element operates in a pipeline stage (all outputs registered)
- local interconnect only (no broadcast)

Semisystolic

- Allows combinational outputs
- Allow broadcast, accumulation
- Retiming algorithms exist for converting to systolic (we will visit when we hit retiming in April)
  - may slow down (create longer paths / cycle)
  - slowdown can be used to solve multiple problems

Systolic Review

- Design style
  - compute limited problems
  - models costs in VLSI systems
- Candidate Problem Requirements
  1. make multiple use of each input data item
     e.g. n inputs $\rightarrow$ computation
     – $O(n \cdot k)$, $O(n^2)$
  2. concurrency
  3. few types of simple cells
  4. simple/regular data and control flow

Non-numeric examples Tuesday.