

Dataflow Architectures

CS533 Spring 2025, UIUC

Jovan Stojkovic

Control-Flow Machines

Control-Flow Machines

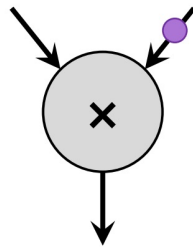
- Conventional architectures are control-flow
 - Sequence of instructions precisely defined
 - Control decides which instruction to execute next via PC
 - If no branches → execute instruction from the next address (e.g., PC+4)
 - If branches → jump to the instruction specified by the branch target

Data-Flow Machines

Data-Flow Machines

- A program consists of dataflow nodes
- A dataflow node fires when all its inputs are ready
 - Fire = Fetch + Execute

▪ Dataflow node (with a **token**) and its ISA representation

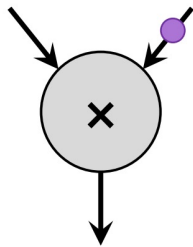


×	0	<i>addr</i>	1	<i>addr</i>	<i>addr</i>
opcode	ready	src	ready	src	dest
	input 1		input 2		

Data-Flow Machines

- How to know that a node's inputs are ready?
- No program counter!

▪ Dataflow node (with a **token**) and its ISA representation



×	0	<i>addr</i>	1	<i>addr</i>	<i>addr</i>
opcode	ready	src	ready	src	dest
	input 1		input 2		

Data-Flow Nodes

- A small set of nodes can define the general programming language

Fork

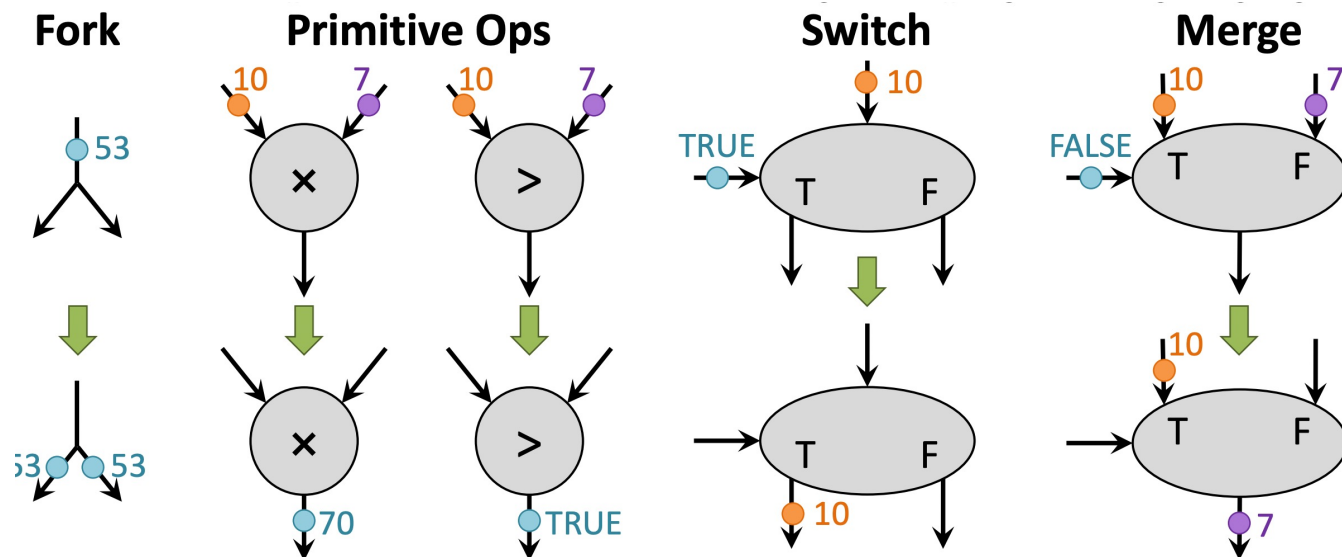
Primitive Ops

Switch

Merge

Data-Flow Nodes

- A small set of nodes can define the general programming language



Flynn's Taxonomy – Where is Data-Flow?

SISD

SIMD

MISD

MIMD

Data-Flow Graph Example

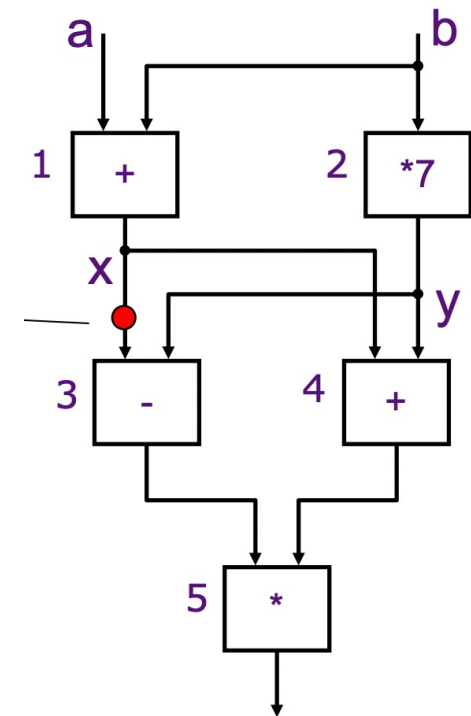
$x = a + b;$

$y = b * 7;$

$\text{return } (x - y) * (x + y);$

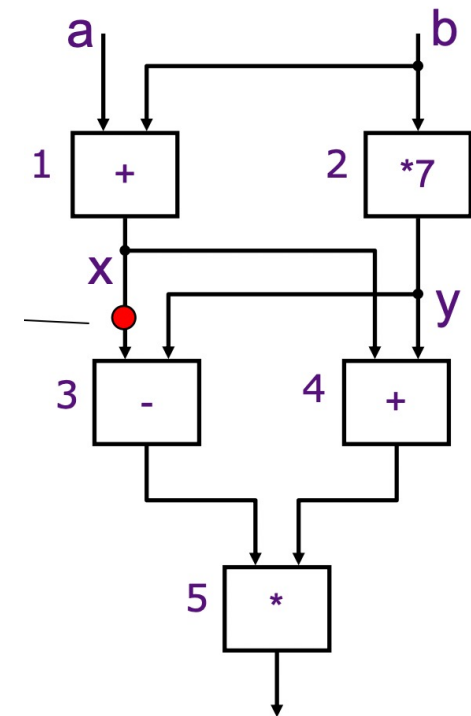
Data-Flow Graph Example

```
x = a + b;  
y = b * 7;  
return (x - y) * (x + y);
```

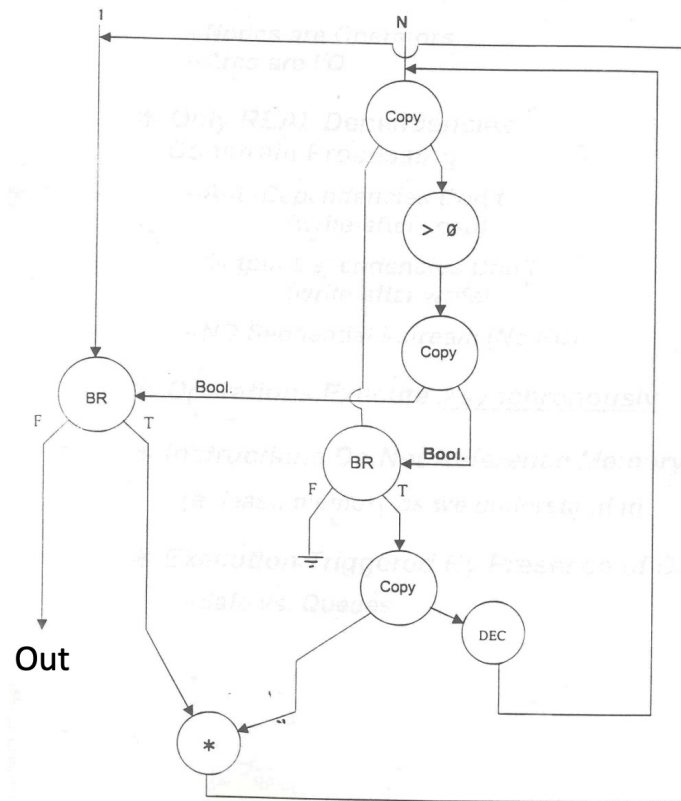


Data-Flow Tokens

- Values in dataflow graphs represented as **tokens**
- Token = <Instruction Pointer, Port, Value>
- An operator executes when all ports ready
- Copy result from the output to the the next input



Data-Flow Example Question?



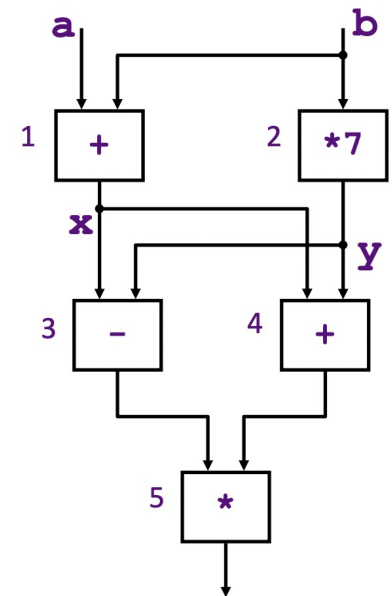
Static Data-Flow Machines

- Allows only one instance of a node to be enabled for firing
- A dataflow node is fired when
 - All input ports ready
 - All output ports empty

	Opcode	Destination 1	Destination 2	Operand 1	Operand 2
1	+	3L	4L		
2	*	3R	4R		
3	-	5L			
4	+	5R			
5	*	out			

Presence bits

Each arc in the graph has an operand slot in the program

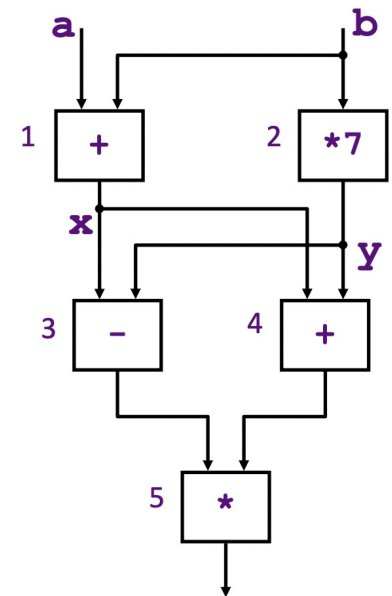


Static Data-Flow Machines

- Allows only one in node to be enabled
- A dataflow node
 - All input ports re
 - All output ports

What are the issues?

Each arc in the graph has an operand slot in the program

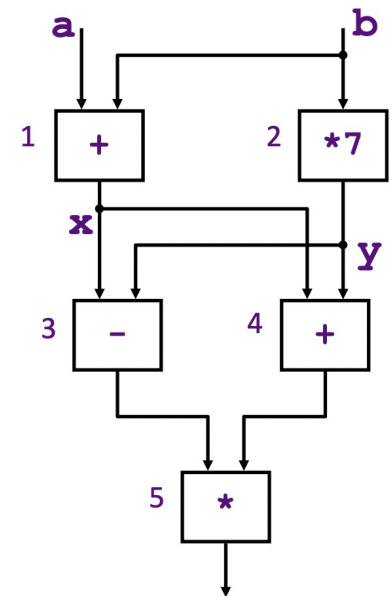


Static Data-Flow Machines

- Allows only one in node to be enabled
- A dataflow node
 - All input ports receive
 - All output ports

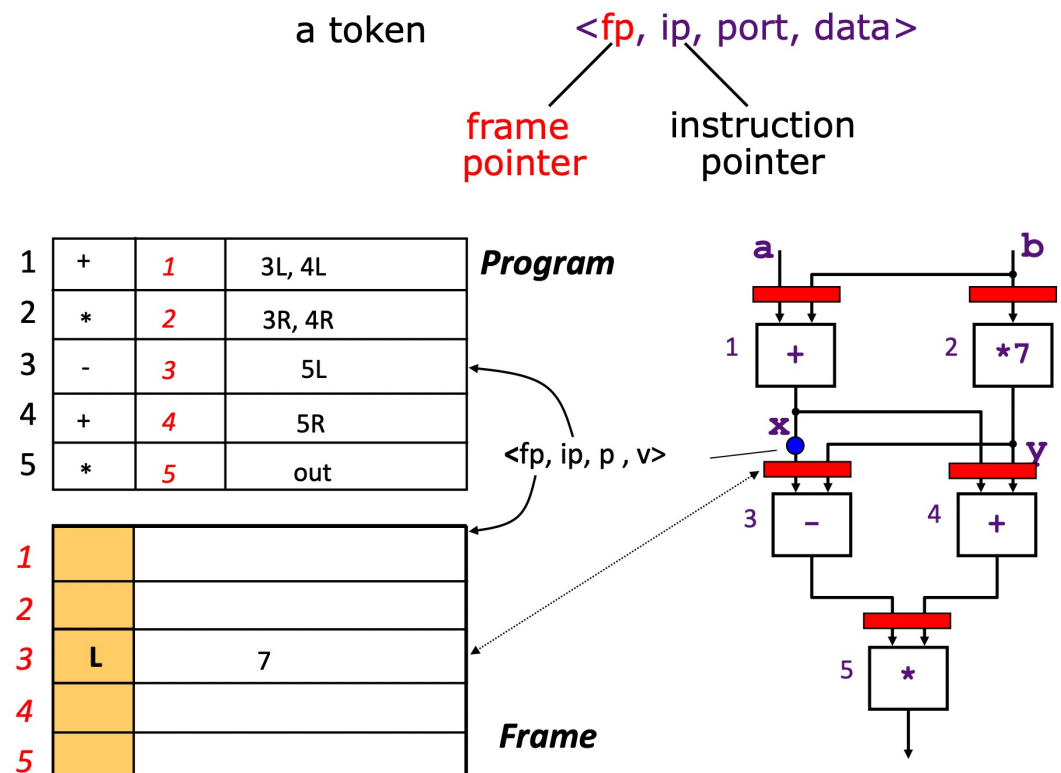
How to solve them?

Each arc in the graph has an operand slot in the program



Dynamic Data-Flow Machines

- Allocate instruction templates (**frames**) dynamically to support each loop iteration and function call
- Code can be shared if we separate code and operand storage



Data-Flow Machines: Pros and Cons

- Advantages

- Disadvantages

Data-Flow Machines: Pros and Cons

- Advantages
 - Can exploit irregular parallelism
 - Only true dependencies constraint processing
- Disadvantages
 - Debugging difficult
 - Implementing complex data structures not easy
 - High bookkeeping overhead
 - Memory locality not exploited