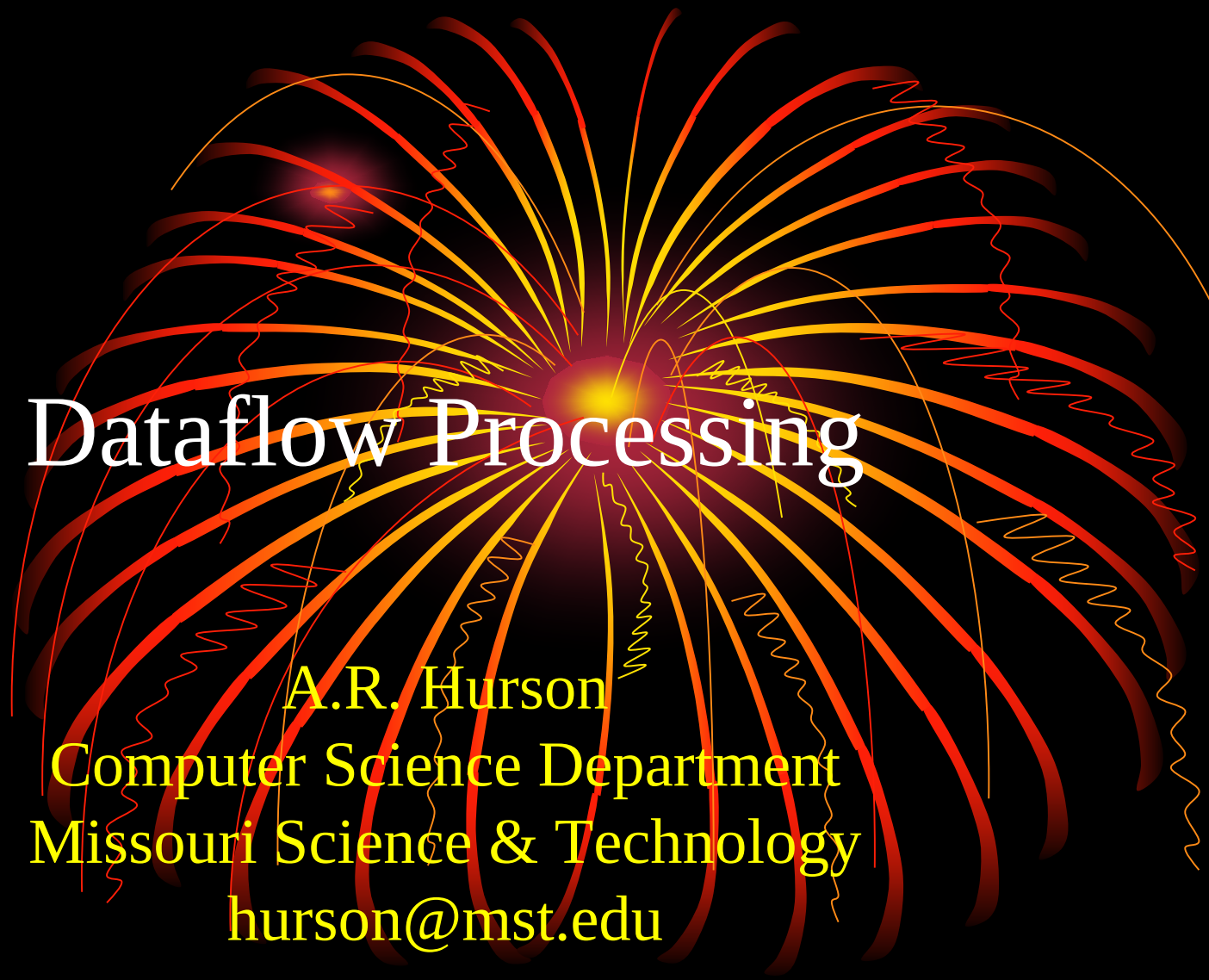


Dataflow Processing

A.R. Hurson
Computer Science Department
Missouri Science & Technology
hurson@mst.edu

Dataflow Processing

- Control Flow Computation
  - Operands are accessed by their addresses.
  - Shared memory cells are the means by which data is passed between instructions.
  - Flow of control is implicitly sequential, but special control instructions can be introduced to explicitly identify concurrency.

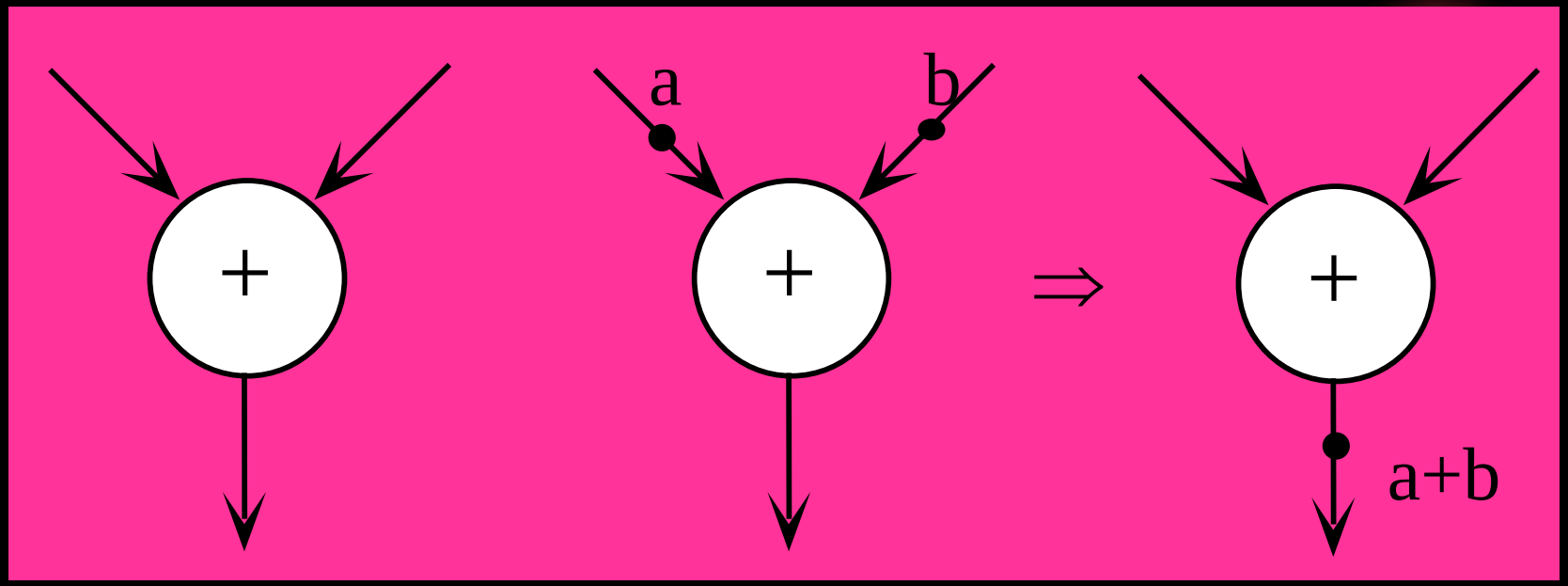
Dataflow Processing

- Control Flow Computation
  - Program Counter(s) is (are) used to sequence the execution of instructions in a centralized environment.

Dataflow Processing

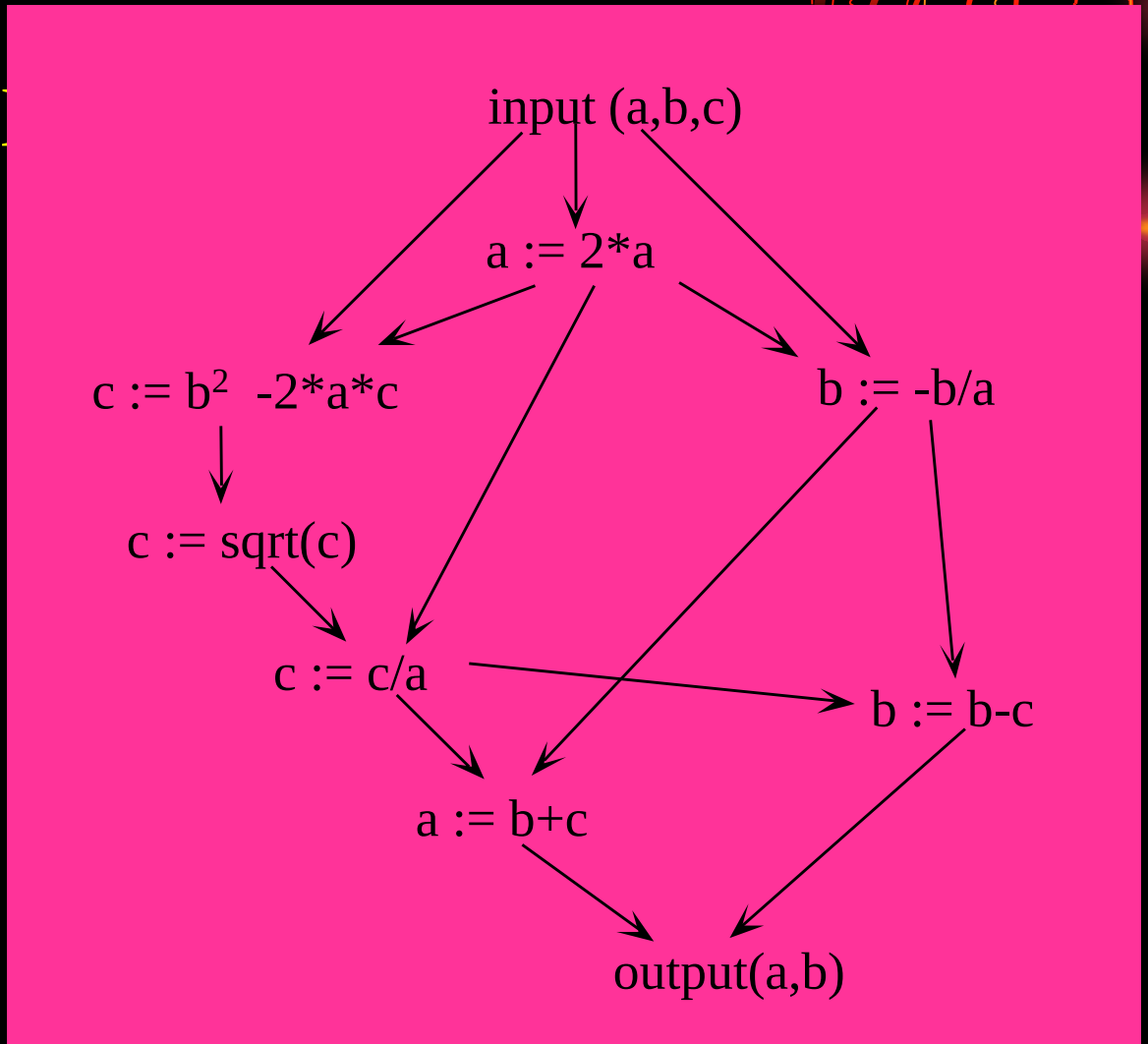
- A dataflow program is a program with a partial ordering defined by the data interdependencies.
- In a dataflow program the activation (execution) of an instruction is triggered (fired) by the availability of its input data.

Dataflow Processing



Dataflow Processing

- Data De



Dataflow Processing

- Dataflow Principles
  - The dataflow model of computation deviates from the conventional control-flow method in two basic principles: asynchrony and functionality:

Dataflow Processing

Dataflow Principles

- Asynchrony: an instruction is fired (executed) only when all the required operands are available.
- Functionality: any two enabled instructions can be executed in either order or concurrently — i.e., no side-effects.

Dataflow Processing

- Dataflow Principles
  - Within the scope of dataflow processing, implicit parallelism is achieved by allowing side-effect free expressions and functions to be evaluated in parallel.

Dataflow Processing

- Dataflow Principles
  - In a dataflow environment, conventional concepts such as variables and memory updating are non-existent.
  - Objects (operand values) are consumed by an actor (instruction) yielding a result object which is passed to the next actor(s).

Dataflow Processing

- Dataflow Principles
  - Within the scope of a concurrent environment, dataflow computation addresses the programmability, memory latency, and synchronization issues.

Dataflow Processing

- Questions
  - Define; programmability, memory latency, and synchronization.
  - How have these issues been addressed in the conventional multiprocessor systems?
  - Why does the dataflow model of computation offer good solutions for these problems?

Dataflow Processing

- Classification
  - The dataflow model of computation has been traditionally classified as either static or dynamic:

Dataflow Processing

- In the static organization, a dataflow actor can be executed only when all of the tokens are available on its input arcs and no token exists on any of its output arcs.
- In the dynamic organization, a dataflow actor can be enabled only when all of the tokens of the same tag (color) are available on its input arcs.

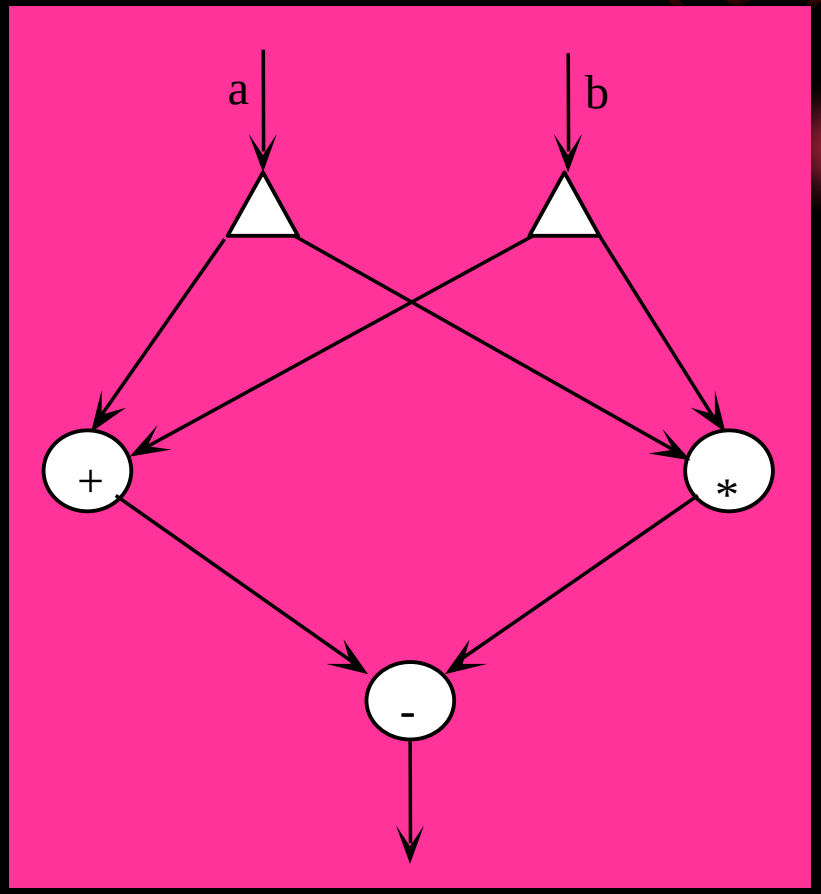
Dataflow Processing

- Dataflow Graph
  - A dataflow program can be represented as a directed graph, G = G(N,A), where nodes (actors) in *N* represent instructions, and arcs in *A* represent data dependencies among the nodes.
  - The operands are conveyed from one node to another in data packets called tokens via the arcs.

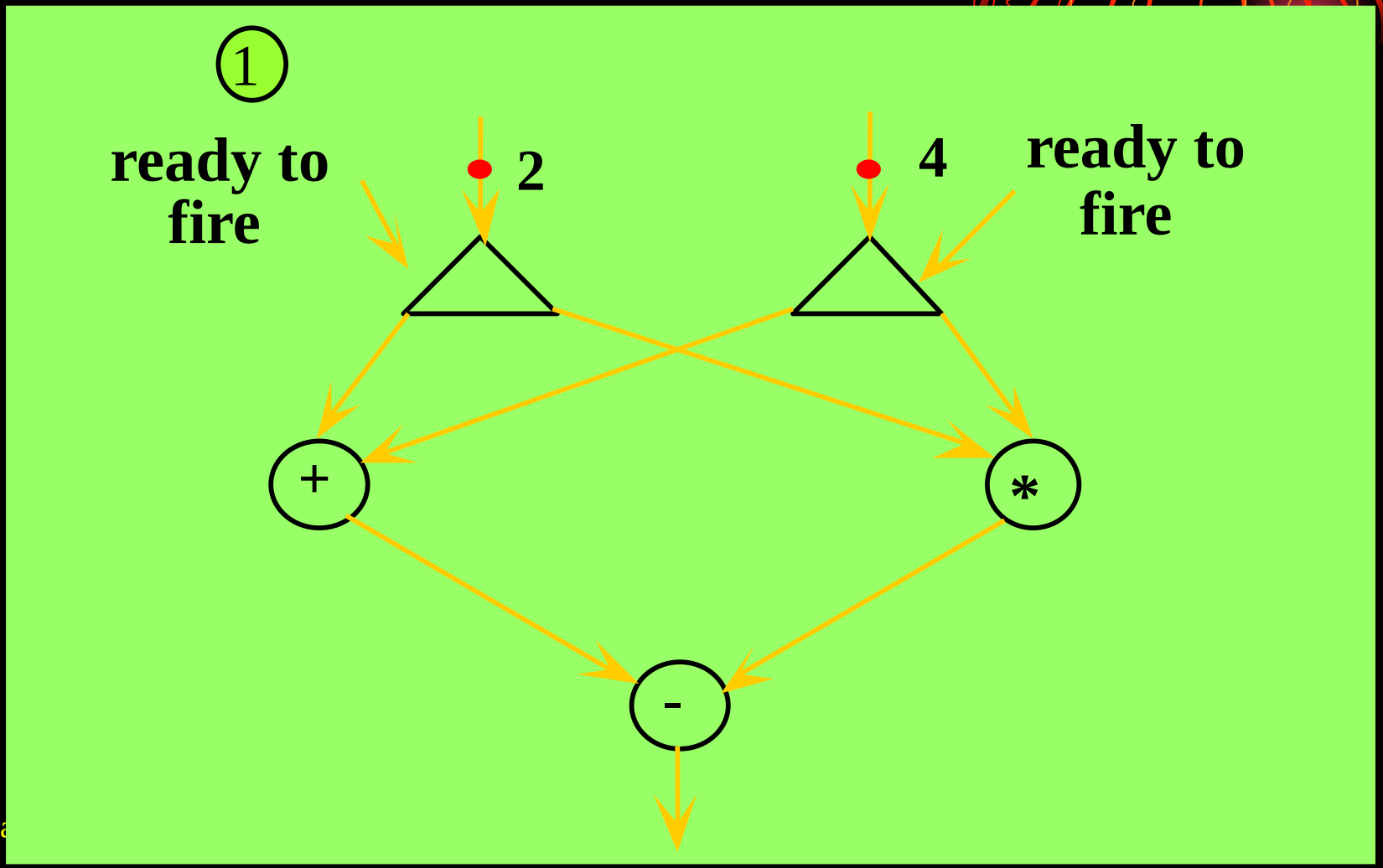
Dataflow Processing

- Dataflow Graph

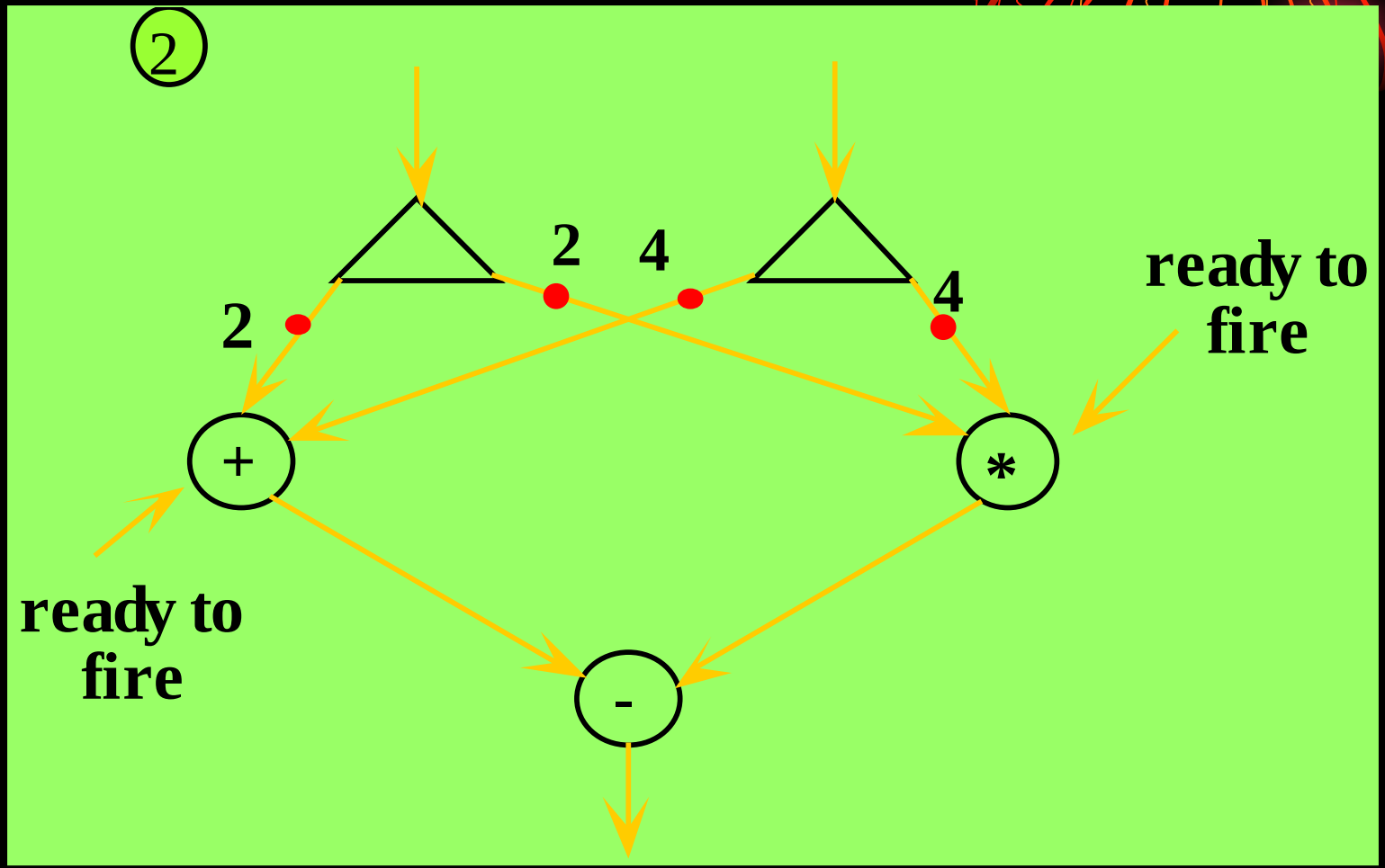
(a+b) - (a*b)



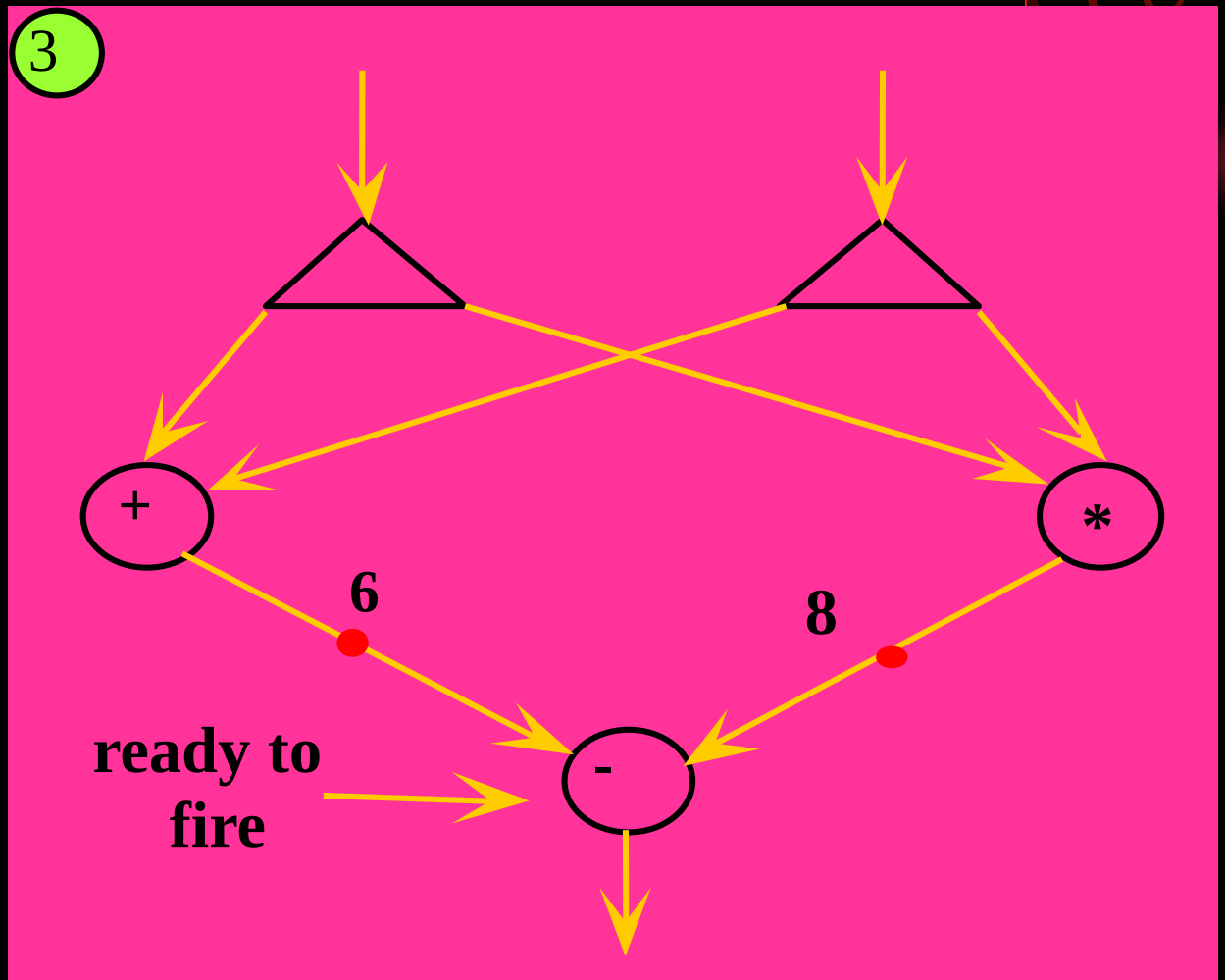
Dataflow Processing



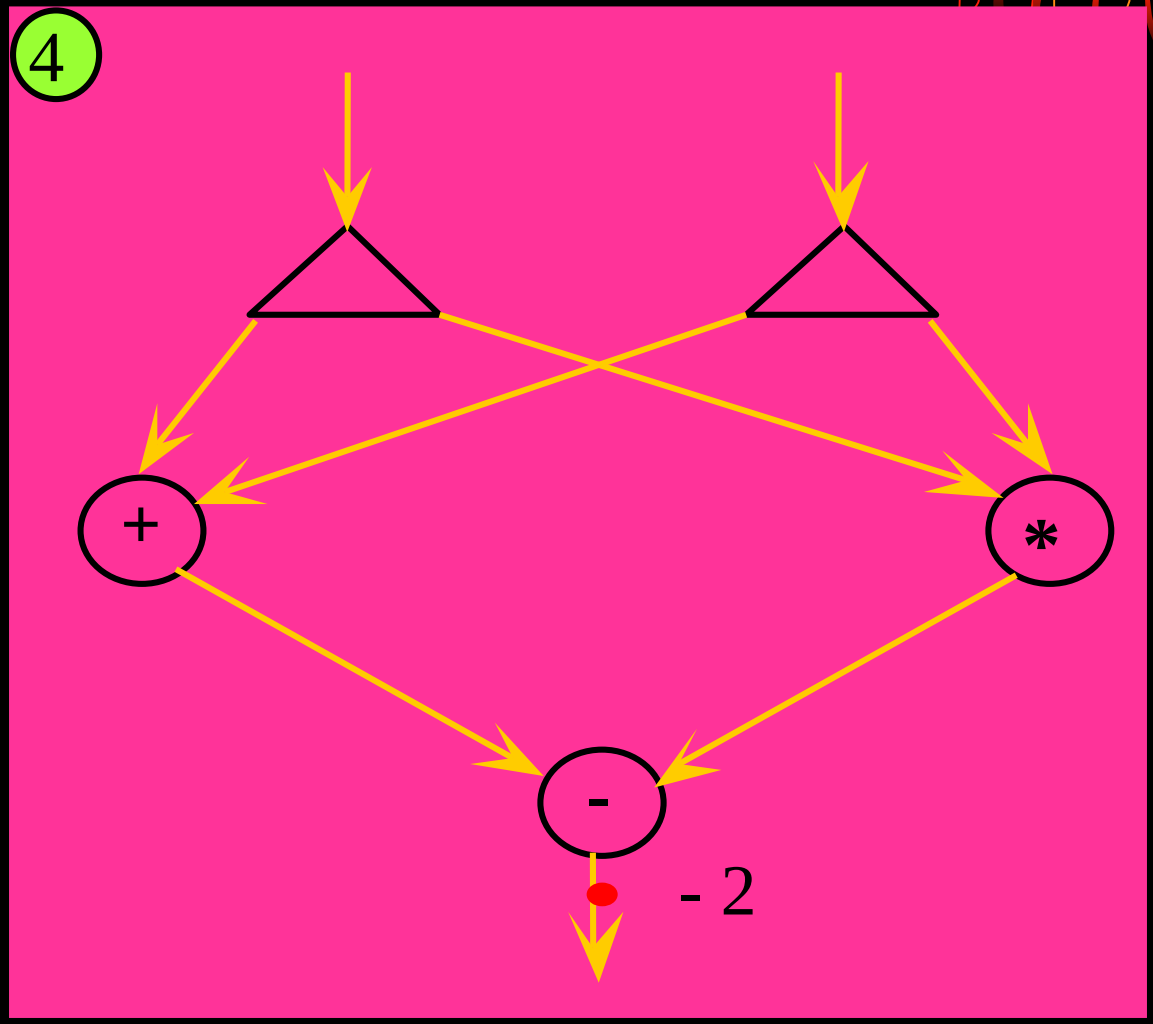
Dataflow Processing



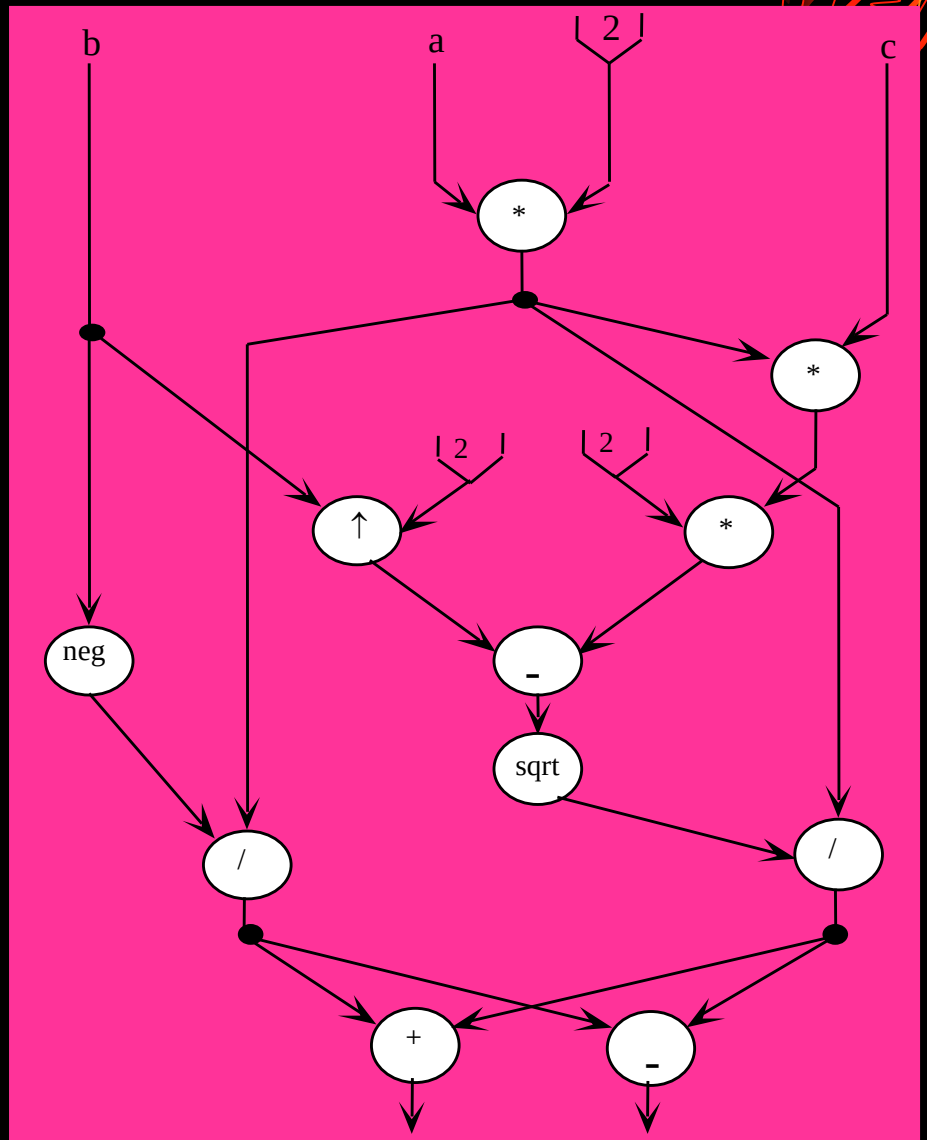
Dataflow Processing



Dataflow Processing



Dataflow Processing



Dataflow Processing

- Dataflow Computation
  - Data are stored in the instructions — i.e., no shared memory.
  - Data are passed among instructions as tokens.
  - An instruction independent of other instructions can begin its execution as soon as it is ready to be fired — e.g., firing rules for static and dynamic environments.

Dataflow Processing

- An Example