

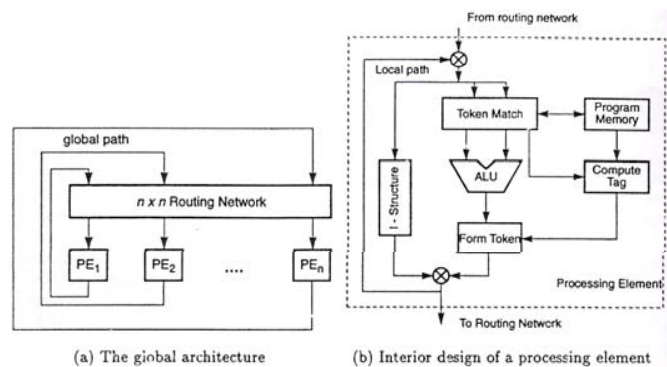
PROGRAM FLOW MECHANISMS

- Control Flow Versus Data Flow
- Demand-Driven Mechanisms
- Comparison of Flow Mechanisms

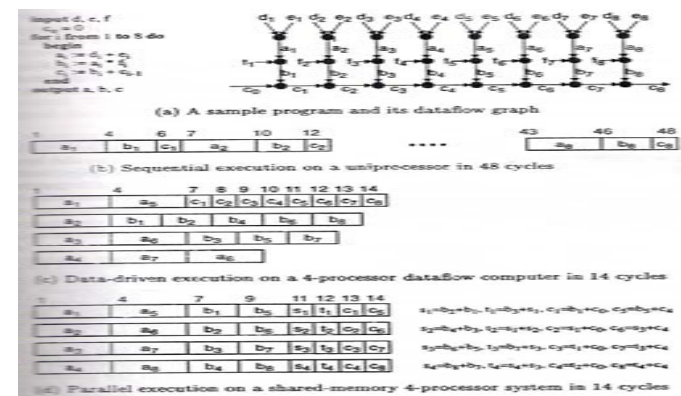
Control Flow Versus Data Flow

- *Control-flow computers* use shared memory to hold program instructions and data objects.
- In a *dataflow computer*, the execution of an instruction is driven by data availability instead of being guided by a program counter.
- Computational results (*data tokens*) are passed directly between instructions.

Dataflow Architecture



Comparison between dataflow and control-flow computers



Demand-Driven Mechanisms

- In a *reduction machine*, the computation is triggered by the demand for an operation's result.
- A demand-driven computation corresponds to *lazy evaluation*, because operations are executed only when their results are required by another instruction.

Reduction Machine Models

- In a *string reduction* model, each demander gets a separate copy of the expression for its own evaluation.
- In a *graph reduction* model, the expression is represented as a directed graph.
- Graph manipulation is based on sharing the arguments using pointers.

Comparison of Flow Mechanisms

Machine Model	Control Flow (control-driven)	Dataflow (data-driven)	Reduction (demand-driven)
Basic Definition	Conventional computation; token of control indicates when a statement should be executed	Eager evaluation; statements are executed when all of their operands are available	Lazy evaluation; statements are executed only when their result is required for another computation
Advantages	Full control	Very high potential for parallelism	Only required instructions are executed
	Complex data and control structures are easily implemented	High throughput Free from side effects	High degree of parallelism Easy manipulation of data structures
Disadvantages	Less efficient	Time lost waiting for unneeded arguments	Does not support sharing of objects with changing local state
	Difficult in programming	High control overhead	Time needed to propagate demand tokens
	Difficult in preventing run-time error	Difficult in manipulating data structures	

(Courtesy of Wah, Lowrie, and Li; reprinted with permission from *Computers for Artificial Intelligence Processing* edited by Wah and Ramamoorthy, Wiley and Sons, Inc., 1990)