

Recap

Level 0 Names to Addresses

Level 1 Address to Values

Level 2 Values to Addresses

Level 3 Memory State Changing Functions

Language Control Structures

eg. If-Then-Else, For Loop,
Repeat-Until, Etc.

VIN Machine Control Units

Program Counter Modification

Programming

What's the purpose of sequencing of Control Flow?
Because we do not want several state-changing functions modifying the same address in unknown order.

We saw this in analyzing the parallelism among statements.
eg. IO, OI, OO dependency.

This is the reason that writing parallel imperative programs is difficult.

Data Flow Paradigm

→ Data Flow Languages

→ Data Flow Programming

→ Data Flow Machines

Basics of Data Flow

Contrast between ViN and Dataflow

① Control-Flow (ViN)

Addressable Variables
(Variables control structure)
causes State-Change

Dataflow

- a) Data is everything.
- b) New values are obtained from given value.

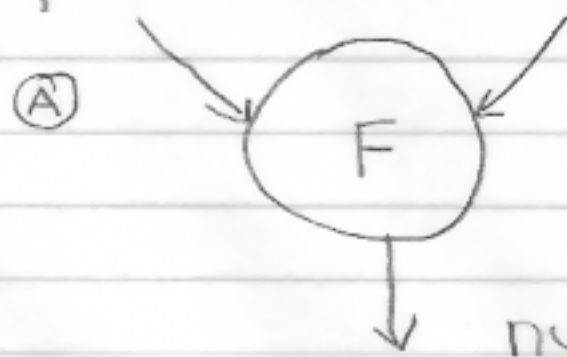
Therefore, we talk about concepts such as:

- arrival of value
- acceptance values
- rejecting values
- c) No concept of PC
- d) A new value is computed (and "output") applying function on the "input" values (values arrived)

Basic Syntax and Semantics of Data Flow

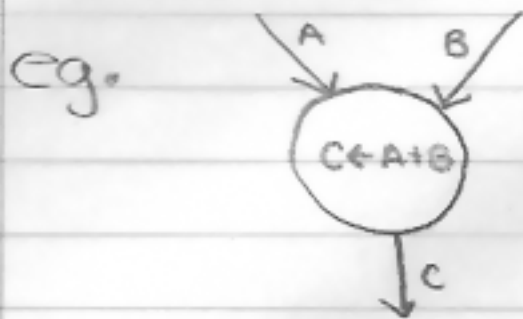
Computation

Syntax

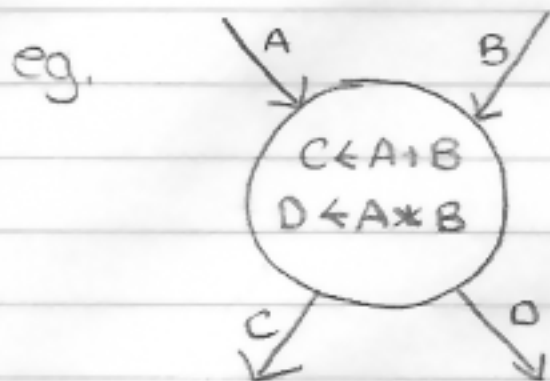


① In Dataflow the program is nothing but a directed graph which has number of input (A, B, etc) lines and a number of output line (eg. C)

Ⓐ The body of a node is a mapping function, defining output (S) in term the input values.

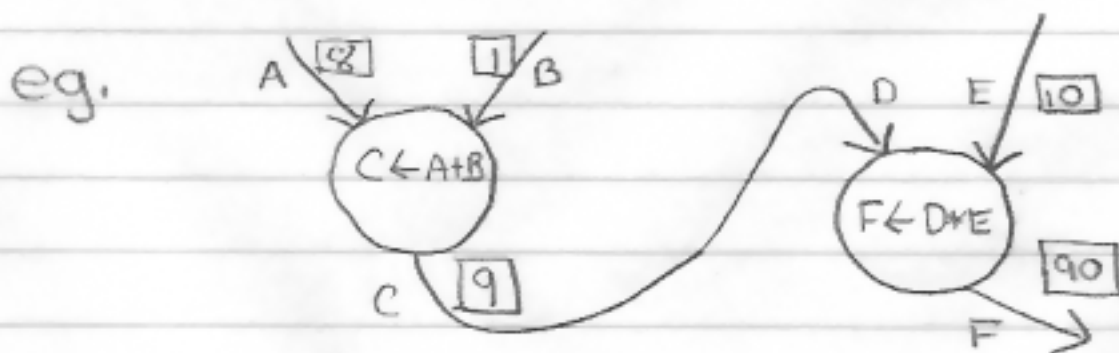


③ The body of a node must have only single-assignment statement



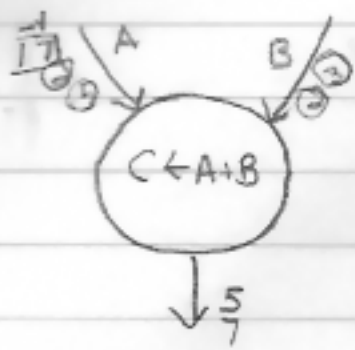
Hence, by virtue of Single-Assignment requirement, it rules out side effects as seen in ViN language.
eg. $A \leftarrow A + B$

④ Composing parts of Dataflow Graph, any output line can be connected to any input line.



∴ Bigger graphs ("programs") can be constructed from smaller graphs ("programs")

Semantics

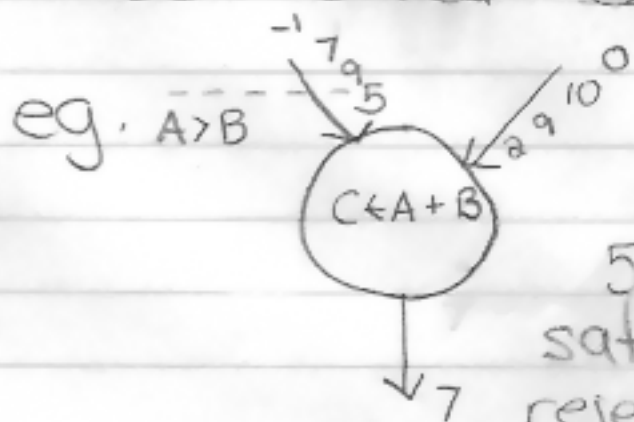


- Any line represents (possibly) a STREAM of Values
 - A node is said to "fire" as soon as there are values in ALL the input lines.
- As a consequence of firing
 - ① It will compute a value (corresponding to the current input values)
 - ② It will put the value on the output line (stream)
 - ③ It "consumes" the set of input values just used

"A Violation of the Basic Dataflow Paradigm"
(Dataflow with context sensitive alter [DFCSF])

Here in DFCSF we can associate a "filter" with a line.

- Filter(s) associated with input line will be called Input Filter
- Filter(s) associated with output line will be called Output Filter



eg. $A > B$

A: 5 9 7 -1

B: 2 9 10 0

$5 > 2 \therefore 7$, $9 > 9$ does not satisfy the filter, thus A is rejected but B is kept.