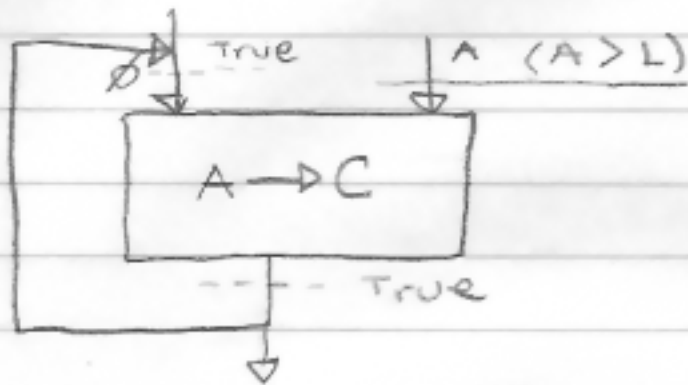


DFCSF (Data Flow With Context Sensitive Filter)

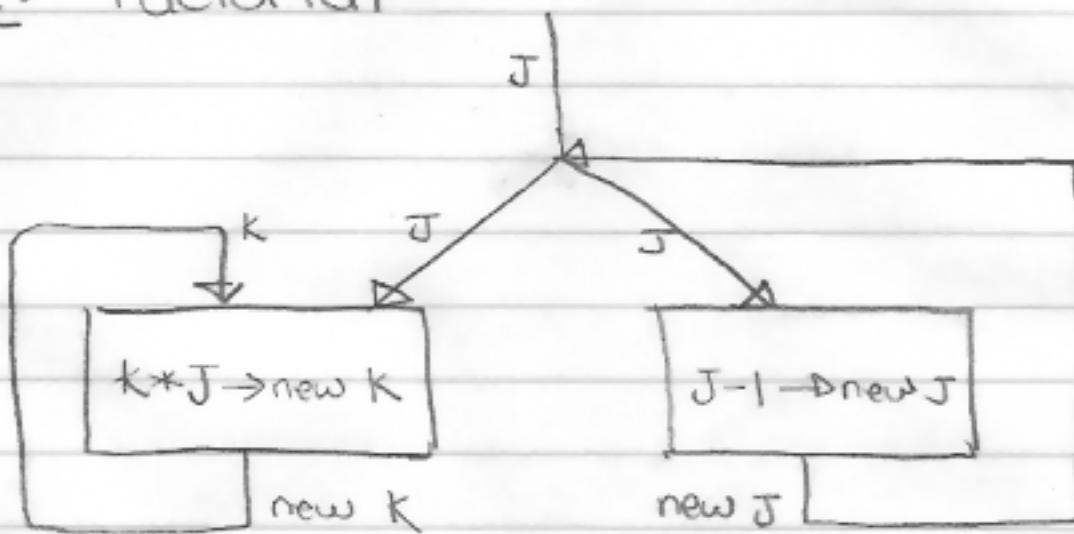
Ex: Computes the largest number so far in A stream A.

L will have the largest number so far.

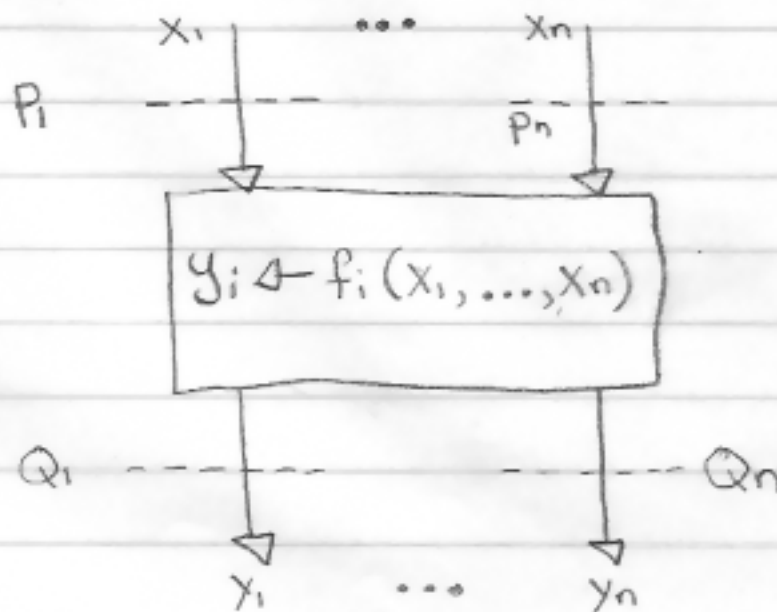


Every I/O you can associate a filter

Ex: Factorial



A typical node in DFCSF



- where X is an input
- P is an input predicate
- Q is an output predicate
- Y is an output

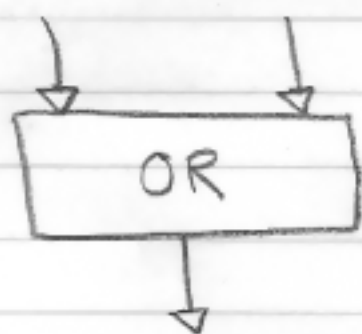
Input predicate uses only the current input values

- If the predicate turns out to be true then it 'retains' the input value, otherwise the value in that line gets rejected. } before the node is fired

Output predicate uses the input values as well as output value.

- If the predicate turns out to be false, then the output value on that line is rejected. } after the node is fired

The other node



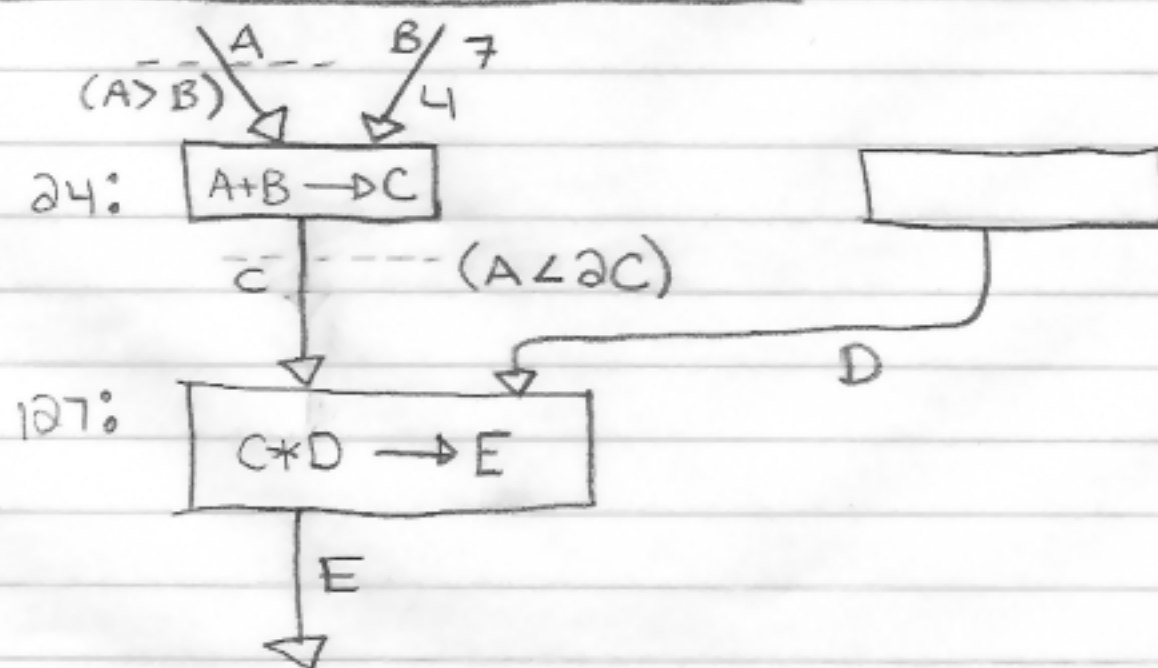
(Nondeterministic OR)

- It serializes (merges) non-deterministically the value on the input line
- Non-strict it does not insist that all the input lines have value.

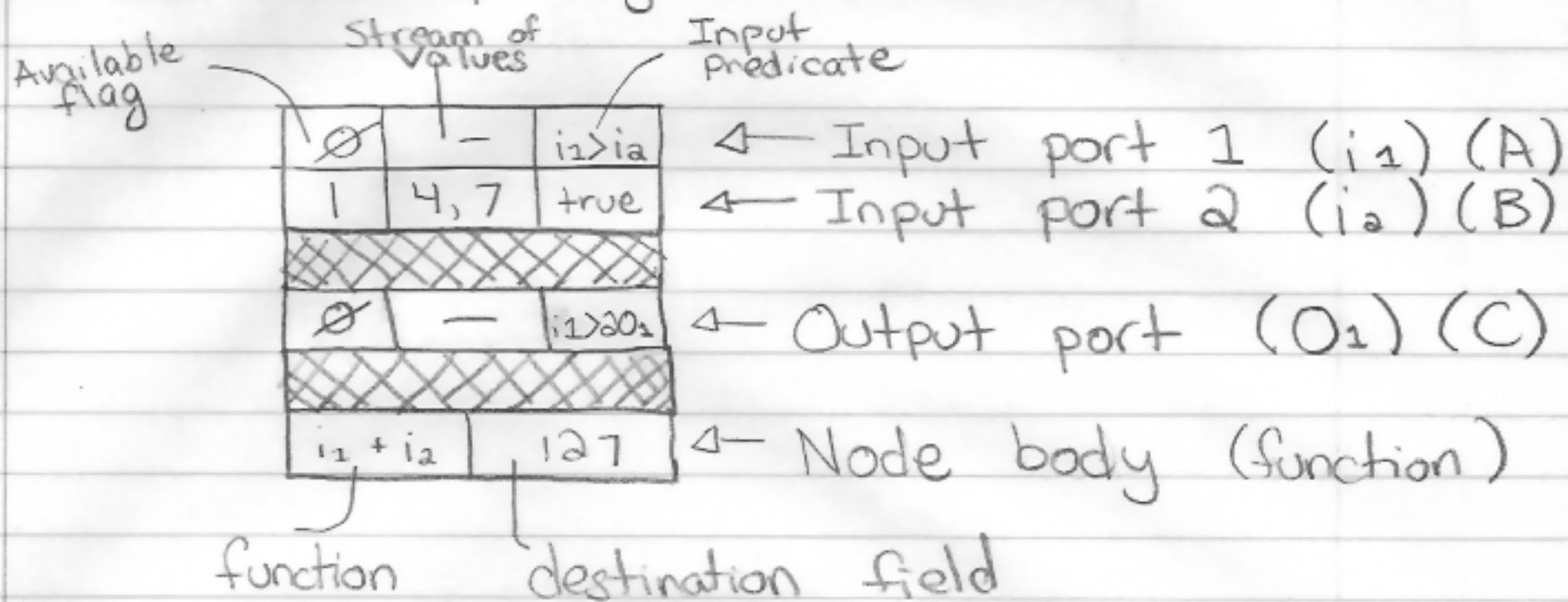
DFCSF Machine

- A cell (I-Cell) for a node.
- Structure of an I-Cell

Ex:



I-Cell Corresponding to no. 24

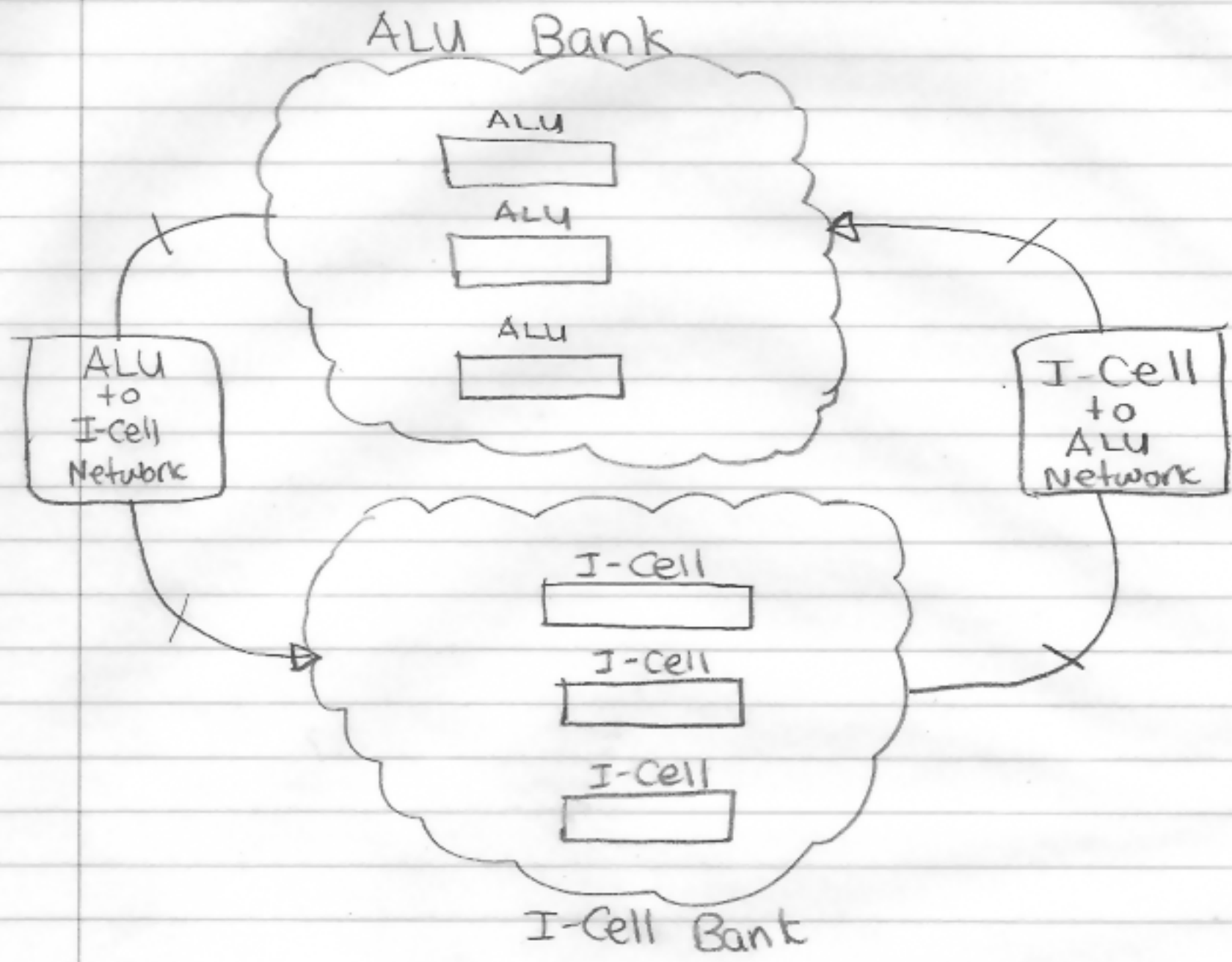


~ Each I-Cell sends a packet to an available ALU

~ An ALU given a packet, will do:

- Computes the input predicates
- Determines if it has all the "necessary" input values
 - ↳ ie. Values on all inputs which satisfy their respective predicate
- And then either "accepts" or "rejects" the current values
- If ALU finds all acceptable inputs then it will compute the output value
- Then it computes the output predicate (using the correct input and output values)
- It absorbs the accepted input that it used. Also it will either accept or reject the output.

Architecture of DFCSE



- Modular
- Scalable

Improvements:

