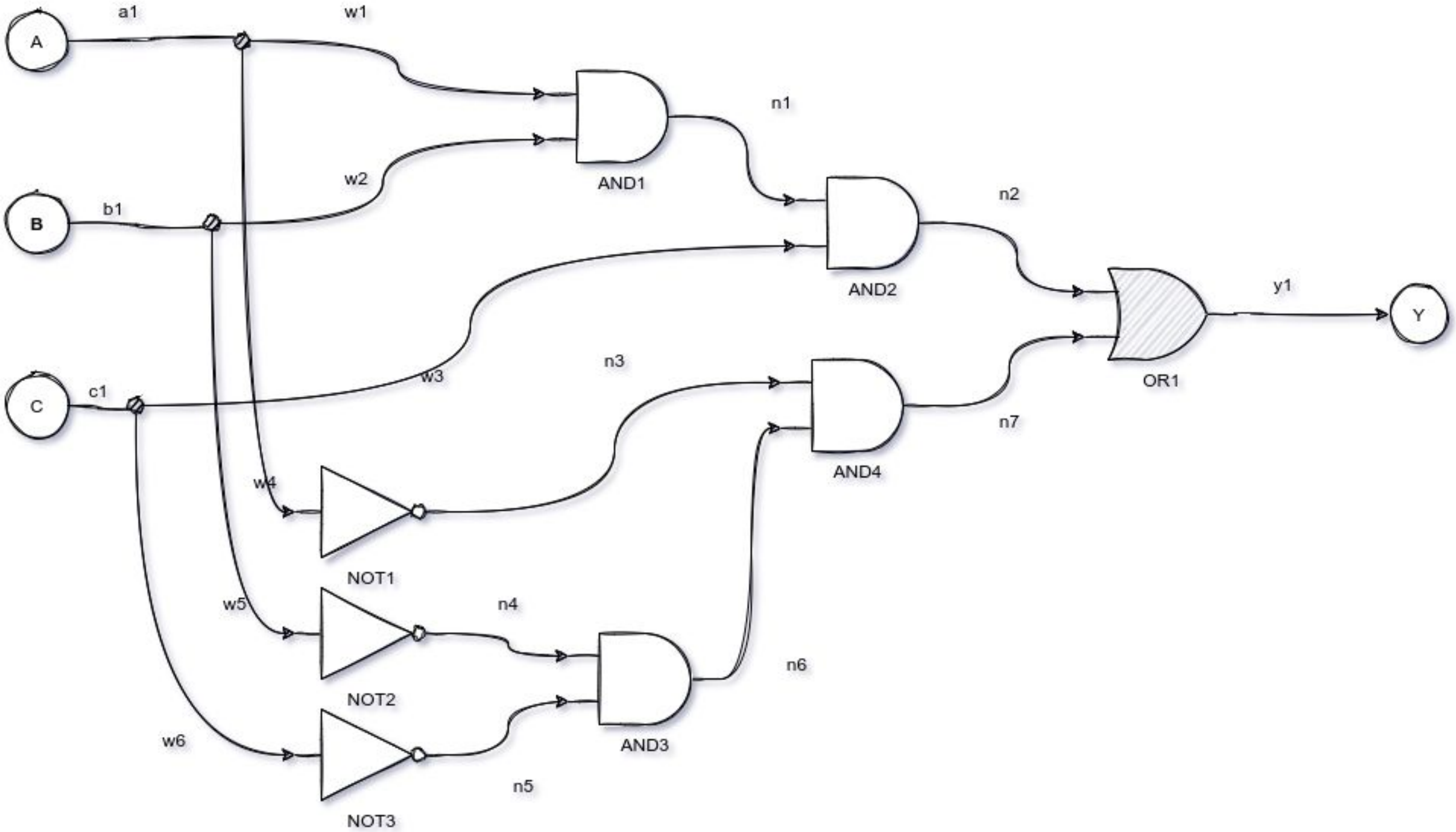
D-Algorithm

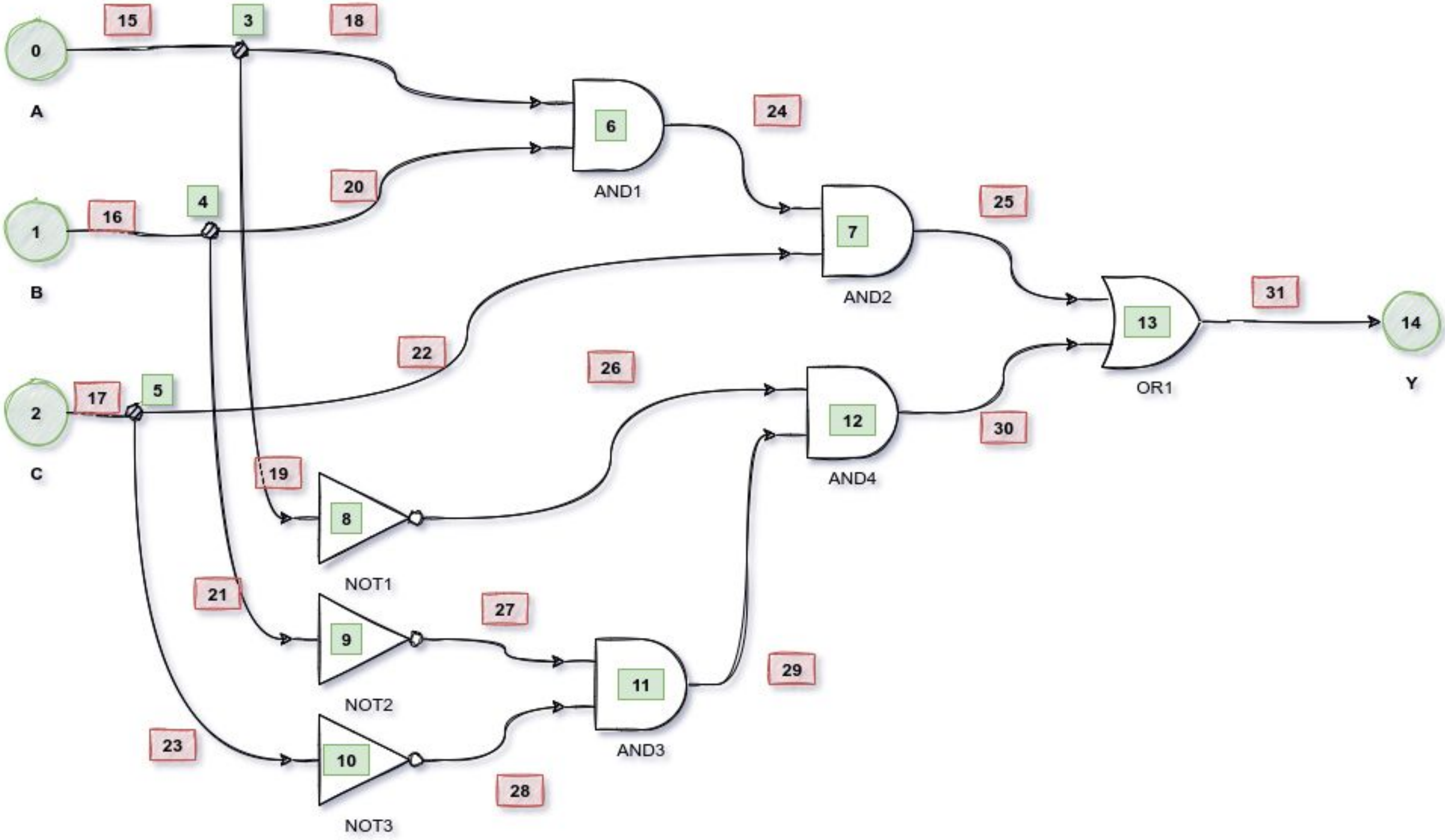
EC804 VLSI Design Testing and Testability

By K. Srikar Siddarth (181EC218)

November 2021

```
# 3 inputs
# 1 outputs
# 8 gates (4 ANDs + 3 NOTs + 1 OR)
# Y = ABC + A'B'C'
# assuming all gates have single output
INPUT A
INPUT B
INPUT C

OUTPUT Y
a1 = fanout A
b1 = fanout B
c1 = fanout C
w1 w4 = fanout a1
w2 w5 = fanout b1
w3 w6 = fanout c1
n1 = AND w1 w2
n2 = AND n1 w3
n3 = NOT w4
n4 = NOT w5
n5 = NOT w6
n6 = AND n4 n5
n7 = AND n6 n3
y1 = OR n2 n7
Y = fanout y1
```


a1
w1
A
AND1
n1
w2
B
b1
AND2
n2
y1
Y
OR1
C
c1
w3
n3
AND4
n7
w4
NOT1
w5
n4
NOT2
AND3
n6
w6
NOT3
n5



15
3
18
0
A
6
AND1
24
20
4
16
1
B
7
AND2
25
22
26
13
OR1
31
14
Y
2
17
5
C
12
AND4
30
19
8
NOT1
21
27
9
NOT2
11
AND3
29
23
10
NOT3
28

Classes

Wire

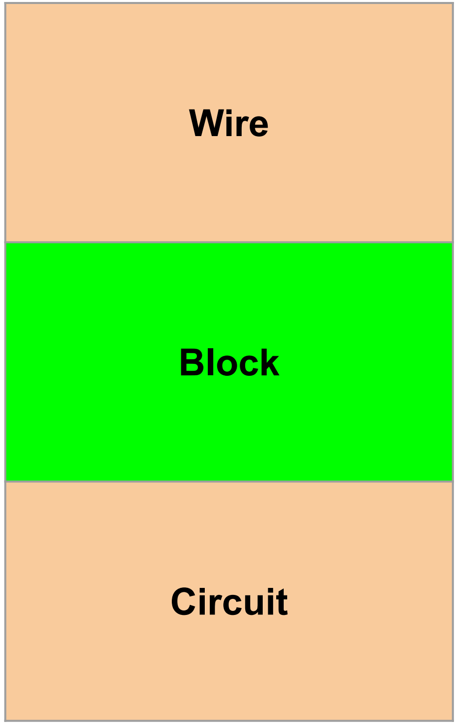
Block

Circuit

Stores the wire objects which lie in between two blocks/gates. For

```
self.type = 'wire'

self.value = 1          # current value

self.inputNode = 3

self.outputNode = 13

self.id = 32

self.name = n2
```

Classes



Stores the Gate object details

self.type = ‘AND’

self.name = ‘AND2’

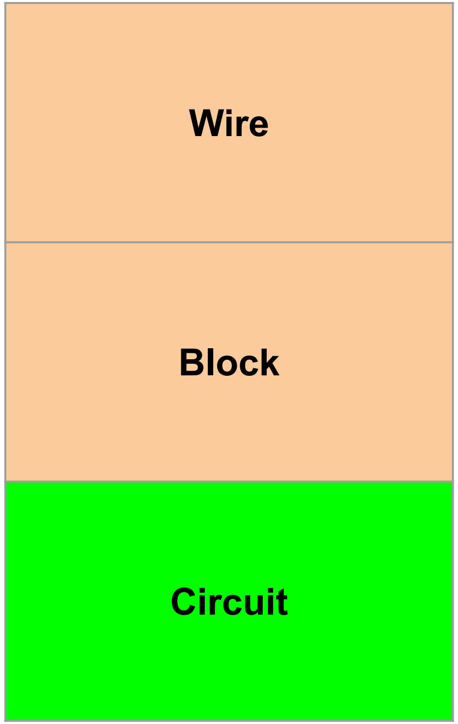
self.id = 7

self.value = 0

self.inputNode = [6,7]

self.outputNode = 13

Classes



Stores the dictionaries for the blocks and wires and includes the functions to implement the D-Algorithm singular cover generator, control signal generator and a few other helper functions and all its necessary variables like the intersection table, D,J Frontiers, Implication Stack.

Main Functions of Circuit class

netlist_to_graph
getGraphFromCircuit
checkConsistency
Imply_and_Check
errorAtPO
getControlValue
inputsAreSpecified
dalg
getCover
save_checkpoint
recover_checkpoint

Converts the input text file into a Python Dictionary

Input : File

Output : Dictionary of Blocks

Main Functions of Circuit class

netlist_to_graph
getGraphFromCircuit
checkConsistency
Imply_and_Check
errorAtPO
getControlValue
inputsAreSpecified
dalg
getCover
save_checkpoint
recover_checkpoint

Adds wires between blocks and completes the process of storing the circuit.

Input : None

Output : Inclusion of Wires in the dictionary

Main Functions of Circuit class

netlist_to_graph
getGraphFromCircuit
checkConsistency
Imply_and_Check
errorAtPO
getControlValue
inputsAreSpecified
dalg
getCover
save_checkpoint
recover_checkpoint

Takes two input values and returns the D - intersection value, which is implemented using a dictionary.

Input : Two Values

Output : Value according to the Intersection table

Main Functions of Circuit class

netlist_to_graph
getGraphFromCircuit
checkConsistency
Imply_and_Check
errorAtPO
getControlValue
inputsAreSpecified
dalg
getCover
save_checkpoint
recover_checkpoint

1. Here we check for the existence of D and J frontiers
2. Apply the assignment queue / implication stack
3. If there are no problems, then update the signal values, otherwise report failure

Input : None

Output : True if there are no inconsistency

Main Functions of Circuit class

netlist_to_graph
getGraphFromCircuit
checkConsistency
Imply_and_Check
errorAtPO
getControlValue
inputsAreSpecified
dalg
getCover
save_checkpoint
recover_checkpoint

Returns a boolean value to check if any error value has been driven to the Primary Output or not.

Input : None

Output : True/False

Main Functions of Circuit class

netlist_to_graph
getGraphFromCircuit
checkConsistency
Imply_and_Check
errorAtPO
getControlValue
inputsAreSpecified
dalg
getCover
save_checkpoint
recover_checkpoint

Returns the control value of a Gate

AND & NAND :- 0

OR & NOR :- 1

Main Functions of Circuit class

netlist_to_graph
getGraphFromCircuit
checkConsistency
Imply_and_Check
errorAtPO
getControlValue
inputsAreSpecified
dalg
getCover
save_checkpoint
recover_checkpoint

Checks if the inputs to any particular gate in unknown. Returns True if all the inputs are initialised to some value or another.

Input : Gate ID

Output : True/False

Main Functions

M. Abramovici, M. A. Breuer, and A. D. Friedman, Digital Systems Testing and Testable Design, Piscataway, New Jersey: IEEE Press, 1994. Revised printing.

netlist_to_graph
getGraphFromCircuit
checkConsistency
Imply_and_Check
errorAtPO
getControlValue
inputsAreSpecified
dalg
getCover
save_checkpoint
recover_checkpoint

Implements the recursive D-Algorithm which employs two lists : D-Frontier and J-Frontier and performs backtracking in case of inconsistencies in the J-Frontier.
The only change made to the algorithm is that while solving the J-Frontiers, instead of assigning the controlling value of the gate to the input nodes, the singular cover of the Gate was assigned.

```
D-alg()
begin
   if Imply_and_check() = FAILURE then return FAILURE
   if (error not at PO) then
     begin
        if D-frontier = ∅ then return FAILURE
        repeat
           begin
              select an untried gate (G) from D-frontier
              c = controlling value of G
              assign c̄ to every input of G with value x
              if D-alg() = SUCCESS then return SUCCESS
           end
        until all gates from D-frontier have been tried
        return FAILURE
     end
   /* error propagated to a PO */
   if J-frontier = ∅ then return SUCCESS
   select a gate (G) from the J-frontier
   c = controlling value of G
   repeat
      begin
         select an input (j) of G with value x
         assign c to j
         if D-alg() = SUCCESS then return SUCCESS
         assign c̄ to j   /* reverse decision */
      end
   until all inputs of G are specified
   return FAILURE
end
```

Main Functions of Circuit class

netlist_to_graph
getGraphFromCircuit
checkConsistency
Imply_and_Check
errorAtPO
getControlValue
inputsAreSpecified
dalg
getCover
save_checkpoint
recover_checkpoint

Returns the set of Inputs for a given Output , i.e, cover of the gate.

Input: Gate with a known output

Output: Input Vectors for that gate

Main Functions of Circuit class

netlist_to_graph
getGraphFromCircuit
checkConsistency
Imply_and_Check
errorAtPO
getControlValue
inputsAreSpecified
dalg
getCover
save_checkpoint
recover_checkpoint

Saves the D-Frontier, J-Frontier, and the Implication Stack before taking decisions during the line justification, so that the previous state can be restored in case any inconsistency occurs with the current decision. Then the algorithm chooses an alternative if available, otherwise it quits.

Main Functions of Circuit class

netlist_to_graph
getGraphFromCircuit
checkConsistency
Imply_and_Check
errorAtPO
getControlValue
inputsAreSpecified
dalg
getCover
save_checkpoint
recover_checkpoint

Recovers the D-Frontier, J-Frontier and the Implication Stack to the earlier saved state.

Results and Disadvantages of D-Algorithm

Fault Trial	Detectable?	Program Output	Expected Output
Sa0 at wire #24: n1	Yes	111	111
Sa1 at wire #24: n1	Yes	001	0x1 / x01
Sa0 at wire #22: w3	Yes	111	111
Sa1 at wire #29: n6	Yes	011	01x / 0x1
Sa1 at wire #31: y1	Yes	011	001 / 010 / 011 / 100 / 101 / 110
Sa1 at wire #15: a1	Yes	1xx	111

Areas of Improvement: Make the algorithm work for errors due to wrong choice of D-Frontier