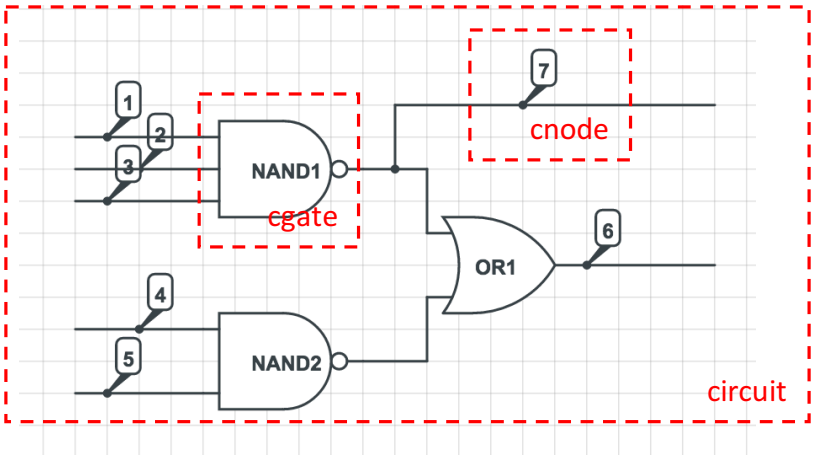
XIAN WU xwu317 903142009

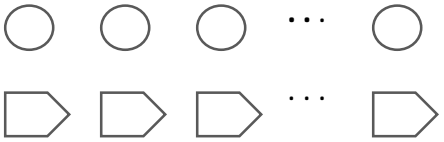
ECE 6140 PROJECT 1#

DATA STRUCTURE



Class circuit

- Vector<cnode> **all_nodes**
- Vector<cgate> **all_gates**
- Vector<int> inputnid
- Vector<int> outputnid



index by node_id

index by gate_id

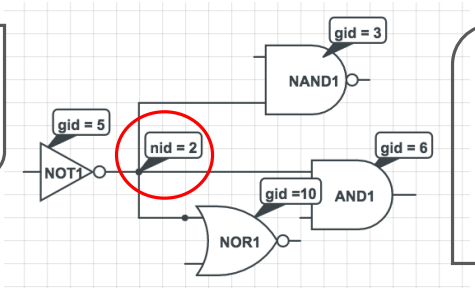
{3 , 6, 10} means node 3 6 and 10 are input nodes

{3 , 6, 10} means node 3 6 and 10 are output nodes

Class cnode

- int nid
- Bool value
- Bool ready

int asoutput : fan-into-it gate id, thus 5



Vector<int>asinput Fan-out-of-it gates id, thus here it is {3,6,10} in the example

Class cgate

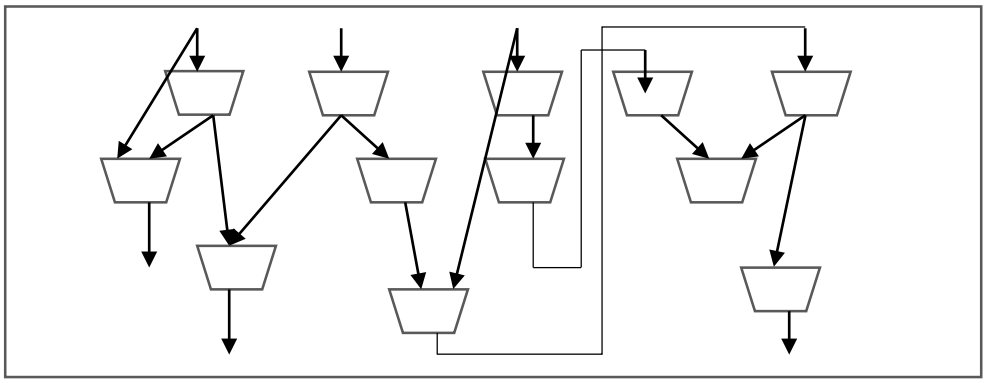
- int gid
- gatetype gtype
- Bool finish

Vector<int> innodes: Input nodes id

int outnodes: output nodes id

ALGORITHM

- Core method is a breadth-first-search to get those outputs



each tree node represents a gate, if some of its inputs are not ready, search stucks there.

"tree node" = "circuit gate" "cnode" = "circuit node"
starts from all inputs, go to gates they connected, if the gate is ready, move on deeper, till at some gate, the search stuck, store the stucked gate id in a vector stuck_gates. After one turn of search, you stuck at some level, then search should starts with those stucked gates. Repeat it till stuck_gates is empty.

- Psedo

```
create a instance of circuit;
add nodes and gates to it;
set ready and set value for input nodes;
declare a vector<int> stuck_gates;

for each input_node:
   add its fan-out-gates to stuck_gates;

while(stuck_gates is not empty):
   if (stuck_gates[i] not ready):
        i+=1; continue;
        //just check next gate, if it still stuck
   else:
        get output value of this gate;
        set the output node ready and set its value;

        add fan-out-gates of this output node to stuck_gates;
        //yes, this means your stuck points are moving forward

        erase stuck_gates[i];

// after get out of the loop, you have already figure out the circuit,
the value of each nodes.

Print output nodes value as the result.
```

Output

Test Circuit	Input	Output
s27.txt	1110101	1001
	0001010	0100
	1010101	1001
	0110111	0001
	1010001	1001
S298f_2.txt	10101010101010101	00000010101000111000
	01011110000000111	00000000011000001000
	11111000001111000	00000000001111010010
	11100001110001100	00000000100100100101
	01111011110000000	11111011110000101101
ss344f_2.txt	101010101010101011111111	101010101010101010101101
	010111100000001110000000	000111100000000100001111100
	111110000011110001111111	000111000001110110001111010
	111000011100011000000000	000011011110011111111000010
	011110111100000001111111	100111011110000010010001000100
s349f_2.txt	101010101010101011111111	101010101010101011010101
	010111100000001110000000	000111100000000101011110000
	111110000011110001111111	000111000001110100011111100
	111000011100011000000000	000011011110011110010001111
	011110111100000001111111	100111011110000010100000100