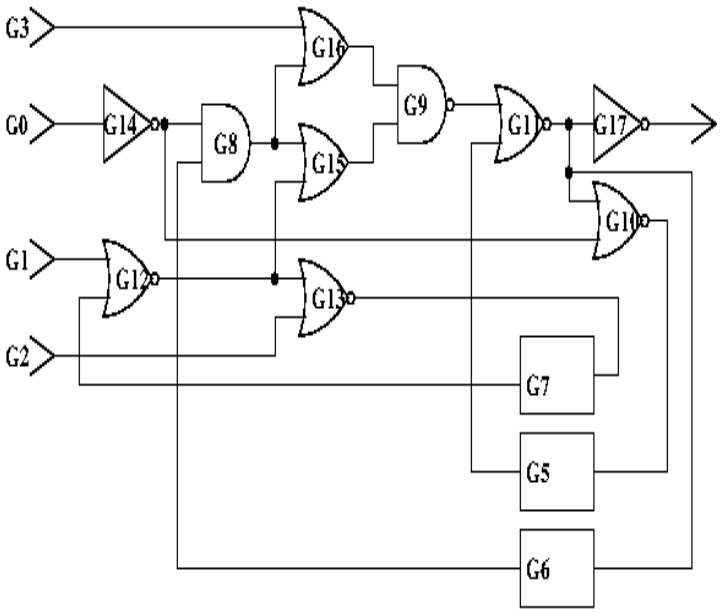
HW5 Due: Nov 24 2015

Figure P1 shows the logic diagram and the corresponding description for a synchronous sequential circuit.



(a) Diagram

```
# 3 D-TYPE FLIP-FLOP
# 2 INVERTERS
# 8 GATES ( 1 AND + 1 NAND + 2 ORs + 4 NORs )
INPUT(G0)
INPUT(G1)
INPUT(G2)
INPUT(G3)

OUTPUT(G17)
G5 = DFF(G10)
G6 = DFF(G11)
G7 = DFF(G13)
G14 = NOT(G0)
G17 = NOT(G11)
G8 = AND(G14, G6)
G15= OR(G12, G8)
G16= OR(G3, G8)
G9 = NAND(G16, G15)
G10 = NOR( G14, G11)
G11 = NOR(G5, G9)
G12 = NOR(G1, G7)
G13 = NOR(G2, G12)
```

(b) Circuit description

Figure P1: ISCAS s27 benchmark circuit

The ISCAS benchmark circuit description, shown in **Figure P1**(b), consists of set of lines. Every line describes how one logic gate is interconnected with other gates. For example, line 'G5 = DFF(G10)' indicates the existence of a D-type Flip-flop with output G5 (connected to the state line q) and input G10 (connected to d). In this description INPUT (G1)/OUTPUT (G17); indicates that line G1 is a primary input/line G17 is a primary output.

a) Write a (C or C++) computer program that reads in the ISCAS circuit description, adds buffers as needed and stores it in the data structure shown in **Figure P2**. To add buffers, for every fanout branch add a buffer gate and modify your data structure to reflect the change. **Figure P3** shows and AND gate with two fanout branches and the AND gate after adding the two buffers. (Add also buffers if needed for output nodes)

Gate name
Gate type
List of pointers to fanin
List of pointer to fanout
Pointer to the next gate

Figure P2: Data record

b) For every gate in the modified circuit associate a 'level'. The gate 'level' indicates the distance of that gate from primary inputs or pseudo inputs (D flip-flop Q's). Initially, the level of primary inputs and DFF flip-flops are set to zero. Gates 'level' for other gates are set to a negative value indicating uninitialized 'level'. Then, the a gate 'level', with assigned positive value on all of its inputs, is equal to the maximum 'level' of its inputs plus one. This step is repeated until every gate 'level' is assigned a positive number.

c) Your program should prints the following:
   1. The total number of gates stored including buffers.
   2. For every 'level' n print the number of gates assigned level n.
   3. Print a listing of the final stored circuit.

Figure P3: Fanout branches