Birla Institute of Technology & Science

I[st] Semester 2015-16

MEL G626 VLSI Test and Testability

Comprehensive Examination (Closed Book)

MM:40 **Time: 3 Hrs** **Date: 01/12/2015 (AN)**

Q1: Write the result displayed after running the following system verilog program.

(i)

```
module PQR;
byte xyz [$];
int j=1;
initial
begin
 xyz.push_front(4);
 xyz.push_front(24);
 xyz.push_front(44);
 xyz.push_back(22);
 xyz.push_back(100);
 $display(" %d ",xyz.size());
 $display(" %d ",xyz.pop_front());
 $display(" %d ",xyz.pop_back());
 xyz.push_back(50);
 xyz.delete(3);
 xyz.insert (3,j);
 $display(" %d ", xyz.size());
 foreach (xyz[i]) $display (xyz[i]);
 end
 endmodule
```

(ii)

```
class packet ;
int P = 0;
function new (int x);
P = x;
endfunction
task print_f();
$display( P );
endtask
endclass
```

```
program main;
initial
begin
packet xyz;
xyz= new(100);
xyz.print_f();
xyz= new(500);
xyz.print_f();
xyz= null;
xyz.print_f();
 end
```

(3+3)M

Q2: For the following code draw the flow chart and find total number of branches, total number of paths and total number of statements.

code:

```
input x,y,z,w,p;
f=1;
e=0;
if (x<0) {
if(y<0)
 e=z*w+1;
else
e=z/w+5; }
else
{
if(y<0)
 f=(z+w)*3;
else
f=(z-w)+5;
}
if(p<0)
 print (f);
else
print (e);
```

Now compute the individual (a) statement/line coverage (i.e. statement ending with semicolon), (b) branch coverage, and (c) path coverage when the following vectors are applied individually.

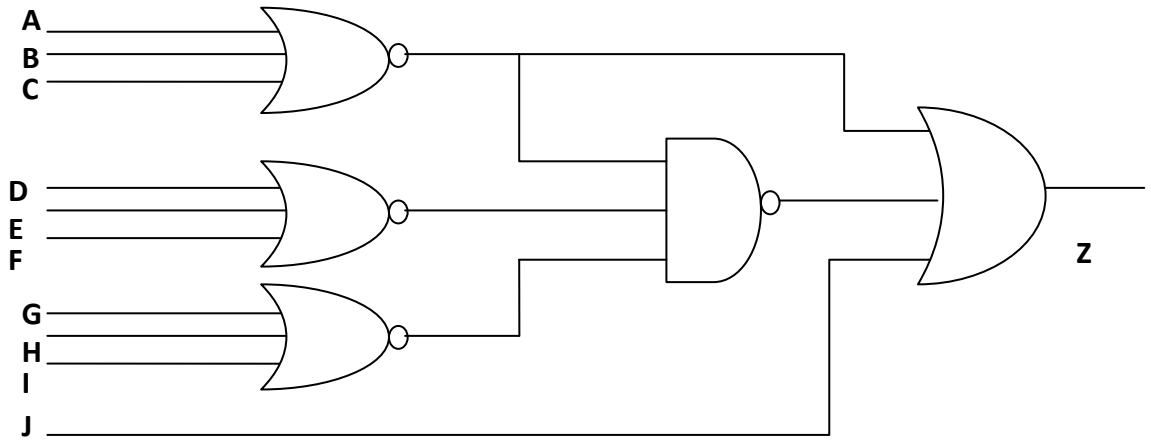
(i) x = -1, y = -1, p = -1

(ii) x = 0, y = 1, p = 1

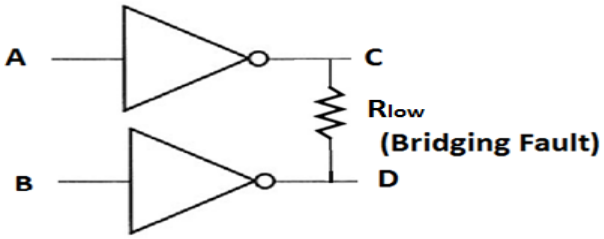
(iii) x = -1, y = 2, p = -1

Also find the accumulated percentage coverage (statement/line, branch coverage, and path coverage) after running all 3 vectors. **(4+3+3)M**

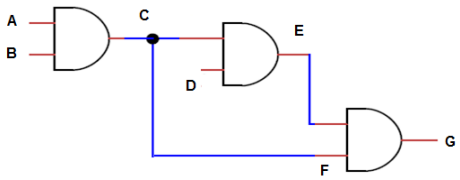
Q3: For the circuit shown below, (i) mark all faults (ii) remove faults using equivalent collapsing and compute the collapse ratio. (iii) remove faults using dominance collapsing and compute the collapse ratio. **(1+2+2)M**



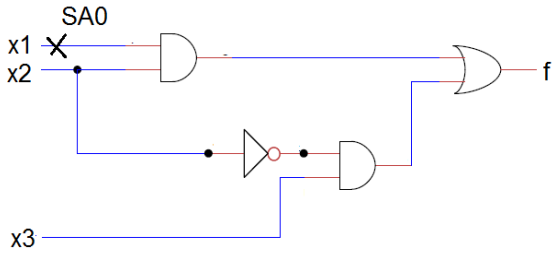
Q4: For the circuit shown below, find the test vector for detecting the bridging fault between 'C' and 'D'. If (R_{An}= R_{Bn}) >(R_{Ap}= R_{Bp}) then also write the type of wired logic required to detect the fault and corresponding faulty and fault free output (Rlow≅0). **(2)M**



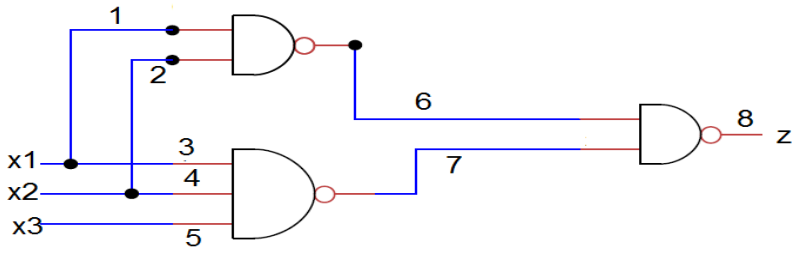
Q5: For the following circuit, Generate the fault list at each node using the deductive fault simulation. ABD={110} **(3)M**



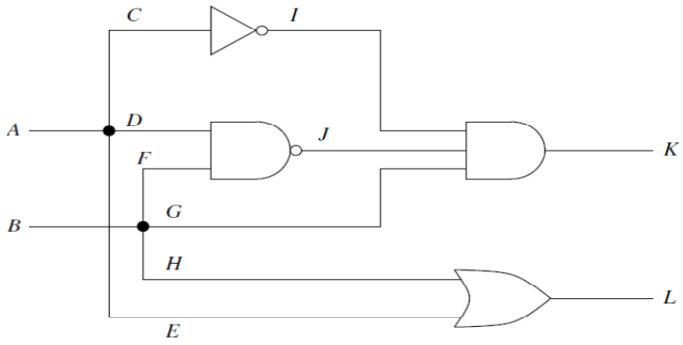
Q6: (A) For the circuit shown below, use the Boolean difference equation to detect SA0 at 'X1'. **(3)M**



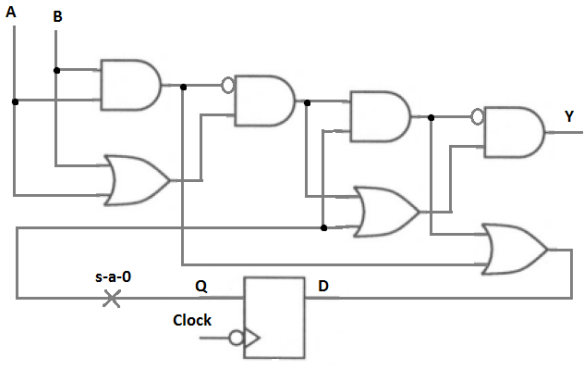
(B) For the circuit shown below use D-algorithm to detect SA0 fault at node '5' ? Use implication procedure using primitive D-cube of failure (PDFC), propagation D-cubes (PDC), singular cover (SC), D-intersection to show values at each note in every steps. **(3)M**



Q7: For the circuit shown below compute SCOAP combinational controllability and observability measures for all lines. Assuming that the testability of a stuck-at fault can be represented as the sum of appropriate controllability and observability, find the set of most difficult to test fault. **(5)M**



Q8: Determine the test sequence for the stuck at zero (s-a-0) fault on the output line of the flip flop in the circuit of figure show below. **(3)M**



----------**********----------