*Name: <u>*Ras Malai*</u>

*Registration: <u>*Apun Bahut Famous Hai!*</u>

Department of Computer Systems Engineering
University of Engineering & Technology Peshawar

Digital System Design

CSE 308

Finalterm Examination Spring 2018

26 June 2018, Duration: 120 Minutes

Exam Rules

Please read carefully before proceeding.

1- This exam is CLOSED books/notes/computers/mobiles.

2- No calculators of any kind are allowed.

3- Answer all problems on the problem sheet.

4- There are 4 problems in total. Some problems are harder than others. Answer the easy ones first to maximize your score.

5- Questions will not be interpreted during the exam.

6- This exam booklet contains 9 pages, including this cover. Count them to be sure you have them all.

Problem 1 —————— (20 pts)

Problem 2 —————— (20 pts)

Problem 3 —————— (15 pts)

Problem 4 —————— (15 pts)

Exam Total —————— (70 pts)

Don't Panic!

Problem 1[CLO-2]: (20 pts) Implement a sequence detector in Verilog using a **behavioral description**. It should be a **Moore FSM**, in which the output is dependent only on the state in which the FSM is and not the current input. The FSM must detect the last two digits of your student ID (i.e. 15-pwcse-dd**dd**), encoded in binary, and **zero extended** to 7 bits. For example, if the last two digits are 89, the FSM should detect 101 1001. If the last two digits are 03, the FSM should detect 000 0011. The FSM output should be high only after the entire sequence has been detected.

Note: Please clearly indicate the sequence that your machine detects. Write a Verilog module which would implement this FSM for input variable "in" and output variable "out." Use the same standard format as was presented in the class. (Define your states; use one always block for next state; use one always block for state transitions and output.)

Solution:

This FSM detects the sequence 101 1001:
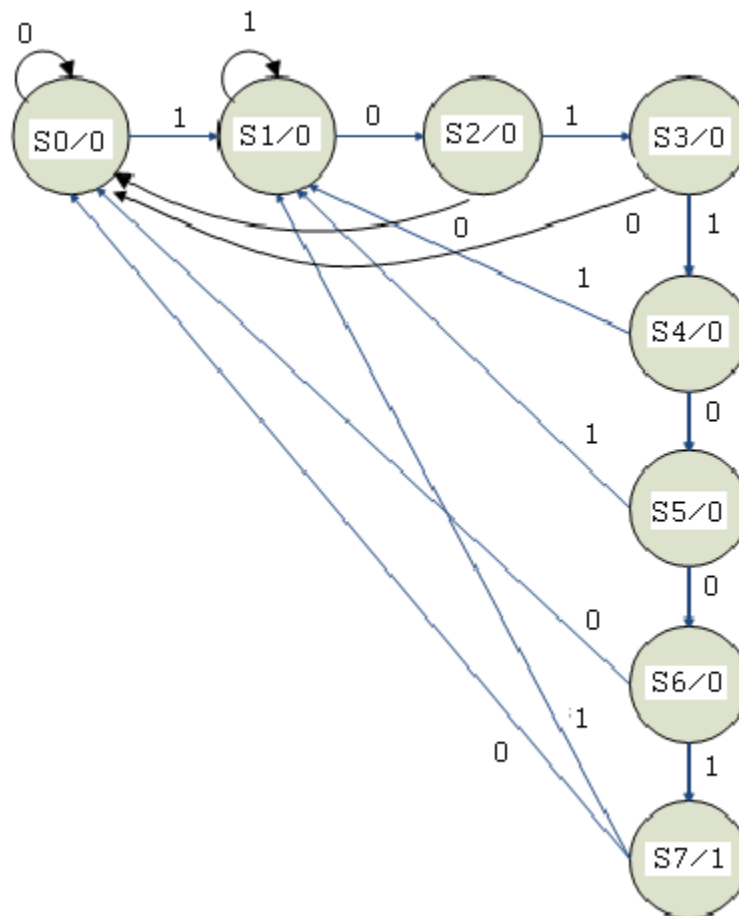
```verilog
module seq_detect(input in, input clk, input reset, output out);

        reg[2:0] curstate, nexstate;

        parameter S0 = 3'b000;
        parameter S1 = 3'b001;
        parameter S2 = 3'b010;
        parameter S3 = 3'b011;
        parameter S4 = 3'b100;
        parameter S5 = 3'b101;
        parameter S6 = 3'b110;
        parameter S7 = 3'b111;

        always @(posedge clk)
                if (reset)  curstate <= S0;
                else        curstate <= nexstate;


        always @(curstate, in)
                case(curstate)
                S0:     begin
                        out = 0;
                        if(in)  nexstate = S1;
                        else    nexstate = S0; end
                S1:     begin
                        out = 0;
                        if(in)  nexstate = S1;
                        else    nexstate = S2; end
                S2:     begin
                        out = 0;
                        if(in)  nexstate = S3;
                        else    nexstate = S0; end
                S3:     begin
                        out = 0;
                        if(in)  nexstate = S4;
                        else    nexstate = S0; end
                S4:     begin
                        out = 0;
                        if(in)  nexstate = S1;
                        else    nexstate = S5; end
                S5:     begin
                        out = 0;
                        if(in)  nexstate = S1;
                        else    nexstate = S6; end
                S6:     begin
                        out = 0;
                        if(in)  nexstate = S7;
                        else    nexstate = S0; end
                S7:     begin
                        out = 1;
                        if(in)  nexstate = S1;
                        else    nexstate = S0; end
                default:        begin
                                out = 0;
                                nexstate = S0; end
                endcase

endmodule
```

Problem 2 [CLO-3]: (20 pts) Consider the circuit in Fig. 1 based on the **single stuck-at fault** assumption and answer the questions related to this.
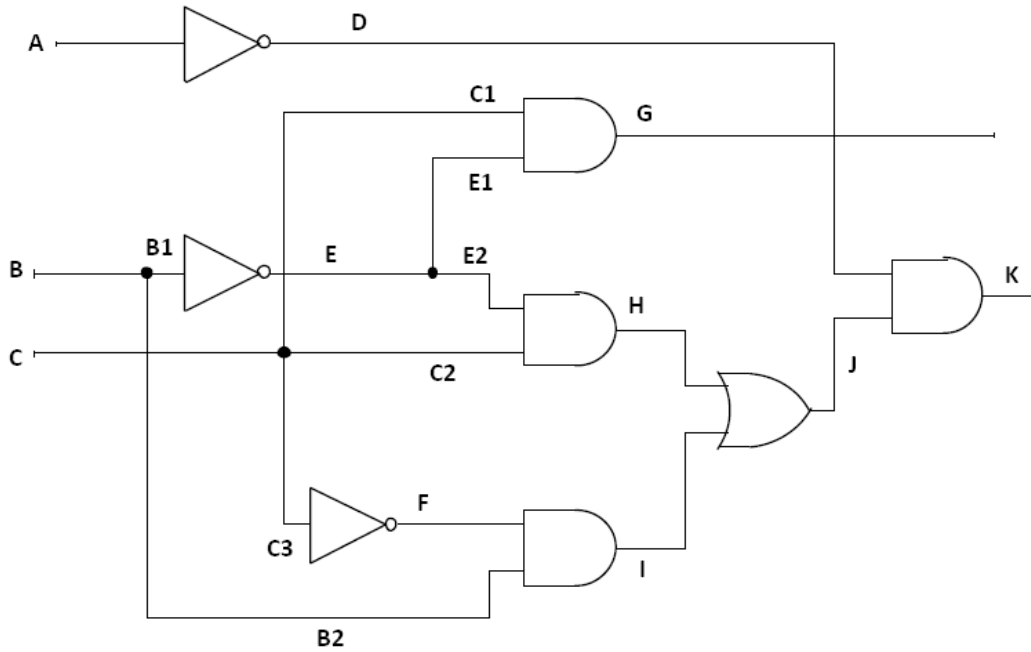

Fig. 1.

(a) (5 pts) What is the number of all possible faults?

(b) (5 pts) Write the reduced fault list using the method of fault equivalence reduction. What is the collapse ratio after you perform fault collapsing (based on the equivalent faults you find)?

(c) (5 pts) Write the reduced fault list using the method of fault dominance reduction. What is the collapse ratio after you perform fault collapsing (based on the dominance faults you find)?

(d) (5pts) How many checkpoint faults are in the circuit? How many of the checkpoint faults can be collapsed by fault equivalence and fault dominance relationships?

Solution:

a) Total faults = 36

b) Fault equivalence collapsing results in 20 faults.

Collapse Ratio (Faults remaining after collapse) = 20/36

X - equivalence

▨ - duplicate

Faults
remaining
after
collapse
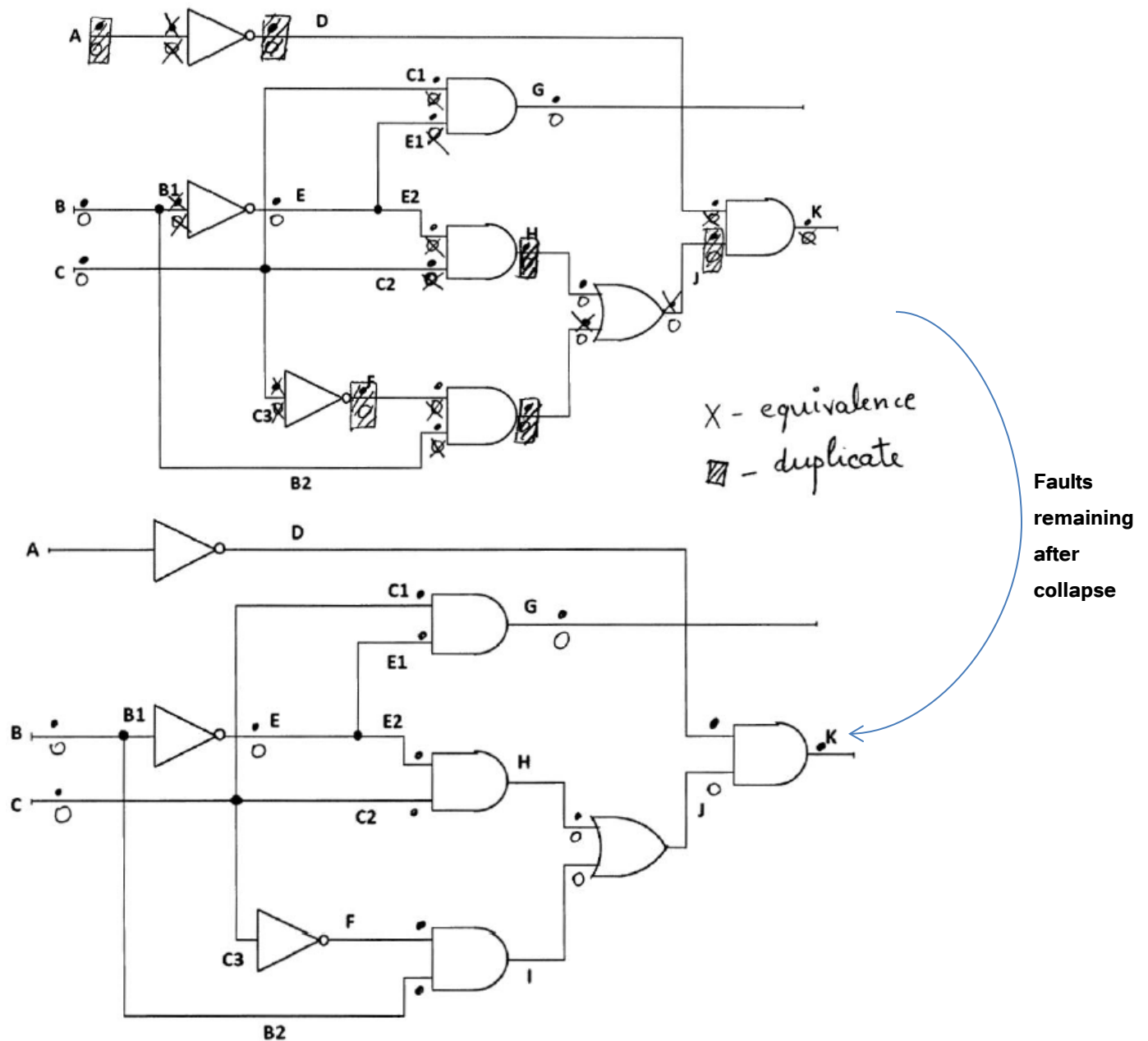
c) Fault dominance is a superset of fault equivalence. Any two faults if equivalent, dominate each other and one of them can be removed. Hence fault dominance should be done on top of fault equivalence.

Fault dominance collapsing results in 16 faults.
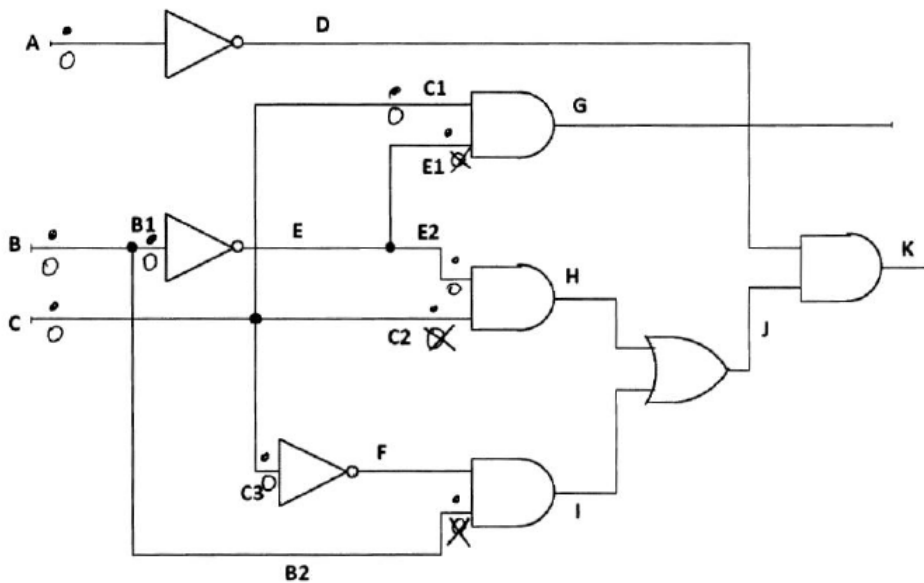Collapse Ratio (after collapse) = 16/36
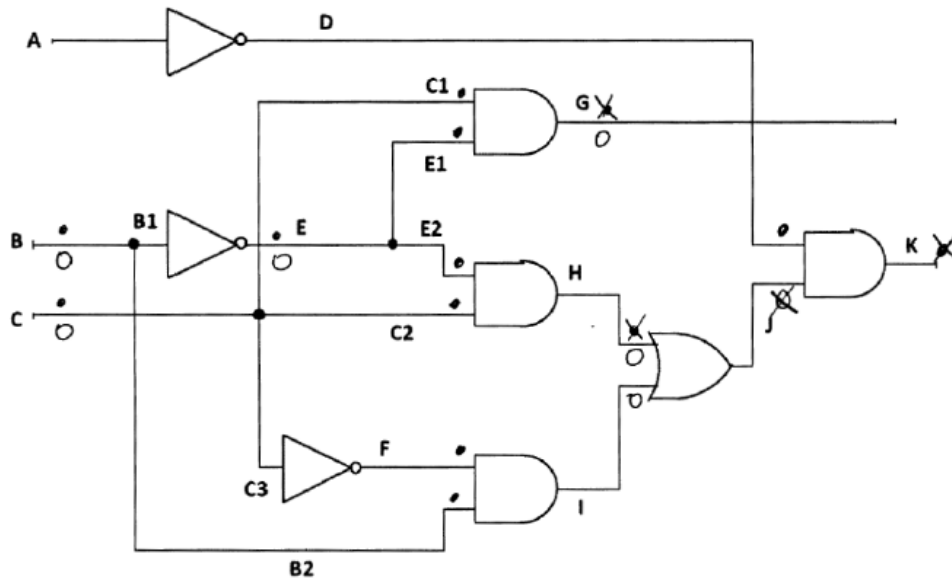
d) There are 20 checkpoint faults.

Equivalence and Dominance should be checked only between these 20 faults. In equivalence and dominance collapsing at least one fault should remain.
The only faults that can be collapsed are E1 s-a-0 or C1 s-a-0 (only one of them), E2 s-a-0 or C2 s-a-0 and B2 s-a-0 or C3 s-a-1.

After collapsing 17 faults remain.

Problem 3 [CLO-3]: (15 pts) Consider the circuit in Fig. 2. Label the sequence of assignments made by the D-algorithm to generate a test for the **s-a-1** fault on line N. Be sure to give the order and show the values for all nodes in the circuit. To simplify the test generation process, whenever there is a choice, you first select the top input. For example, to justify M=1 in Fig. 2, you first justify I=1.
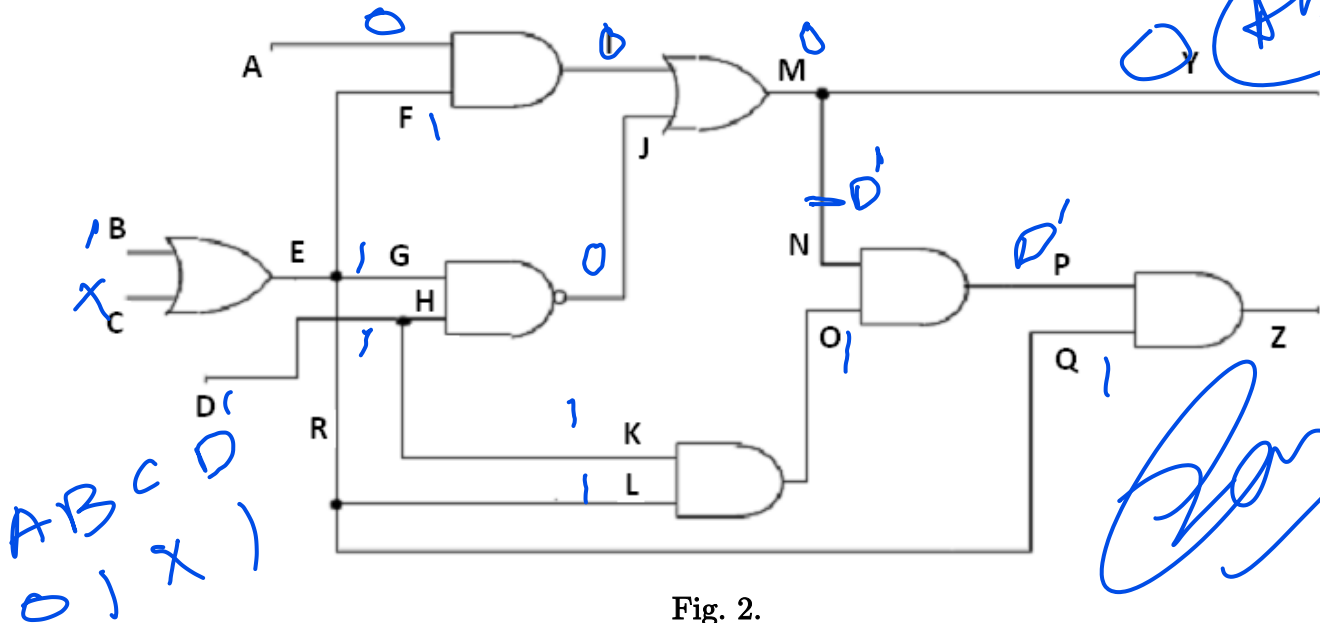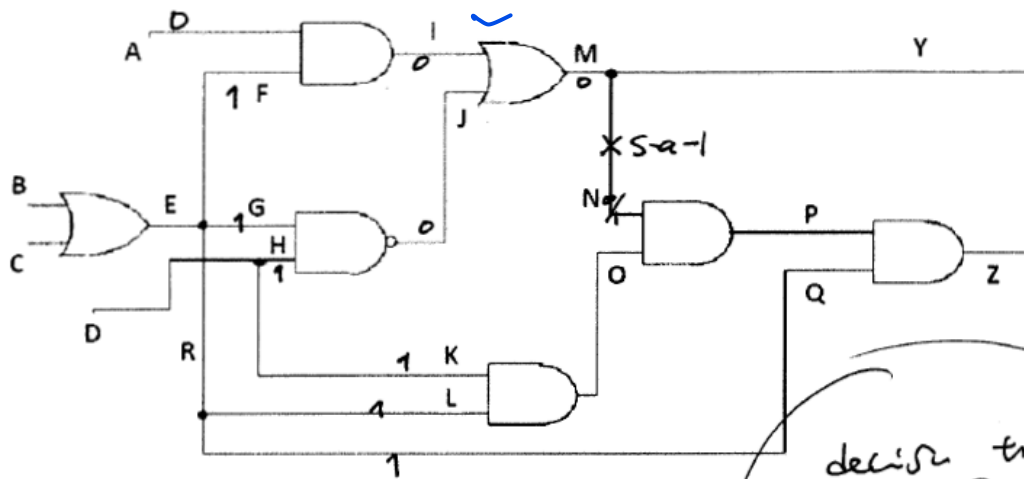

Fig. 2.

Solution:

Fig. 1

D-algorithm :

Decision	implication	Comment
	M = 0, I=0, J=0	implied by (N)
	G=1, H=1, F=1, D=1	fault activate &
	A=0, K=1, L=1	fault propag (O)
	R=1, Q=1, O=1,	
	P = ½, Z=½	J-front : $\left(E = 1 \right)$
B=1	E = 1	

Test pattern: A=0, B=1, D=1, no backtrack for decision tree.

Problem 4 [CLO-3]: (15 pts) Using the circuit shown in Fig. 3, compute the set of all vectors that can detect each of the following faults using the Boolean difference.
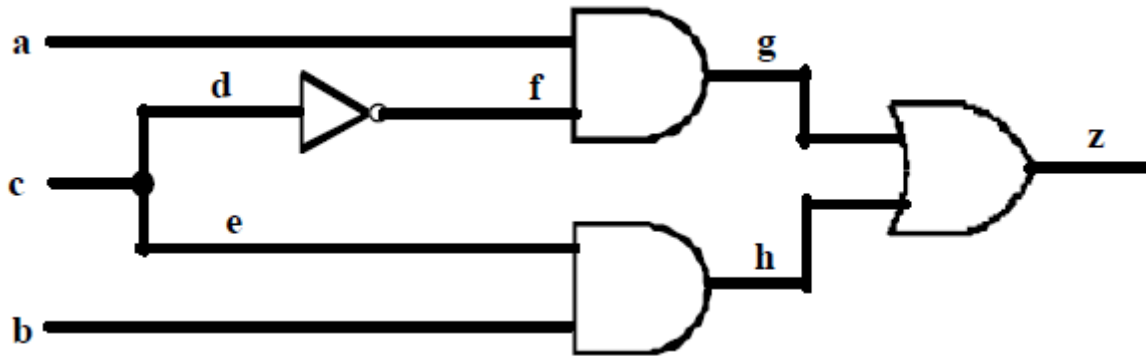
(a) (5pts) e s-a-0
(b) (5pts) e s-a-1
(c) (5pts) c s-a-0

Fig. 3.

Solution:

(a) The set of vectors that can detect e/0 is

$$(e=1) \cdot \frac{d_z}{d_e} = e \cdot (z \ (e=0) \oplus z(e=1))$$

$$= e \cdot (g \oplus (g+b))$$

$$= e \cdot \left(g \cdot \overline{(g+b)} + \overline{g} \cdot (g+b) \right)$$

$$= e \cdot \overline{g} \cdot b$$

$$= e \cdot \overline{\overline{ac}} \cdot b$$

$$= e \cdot \left(\overline{a} + c \right) \cdot b$$

$$= bc + \overline{a}bc$$

$$= bc$$

Thus, the set of vectors is $\{011, 111\}$.

(b) The set of vectors that can detect e/1 is

$$(e = 0) \cdot \frac{d_z}{d_e} = \overline{e} \cdot (z \ (e = 0) \oplus z(e = 1))$$

$$= \overline{e} \cdot (g \oplus (g + b))$$

$$= \overline{e} \cdot \left(g \cdot \overline{(g + b)} + \overline{g} \cdot (g + b) \right)$$

$$= \overline{e} \cdot \overline{g} \cdot b$$

$$= \overline{e} \cdot \overline{ac} \cdot b$$

$$= \overline{e} \cdot \left(\overline{a} + \overline{c} \right) \cdot b$$

$$= b\overline{c}\overline{e} + \overline{a}b\overline{e}$$

$$= \overline{a}b\overline{e}$$

$$= \overline{a}b\overline{c}$$

Thus, the set of vectors is {010}.

(c) The set of vectors that can detect c/0 is

$$(c = 1) \cdot \frac{d_z}{dc} = e \cdot (z \ (c = 0) \oplus z(c = 1))$$

$$= c \cdot (a \oplus b)$$

$$= c \cdot \left(a \cdot \overline{b} + \overline{a} \cdot b \right)$$

$$= a\overline{b}c + \overline{a}bc$$

Thus, the set of vectors is {011, 101}.