NCTU-CS Digital System Lab.

Online Test (11/27)

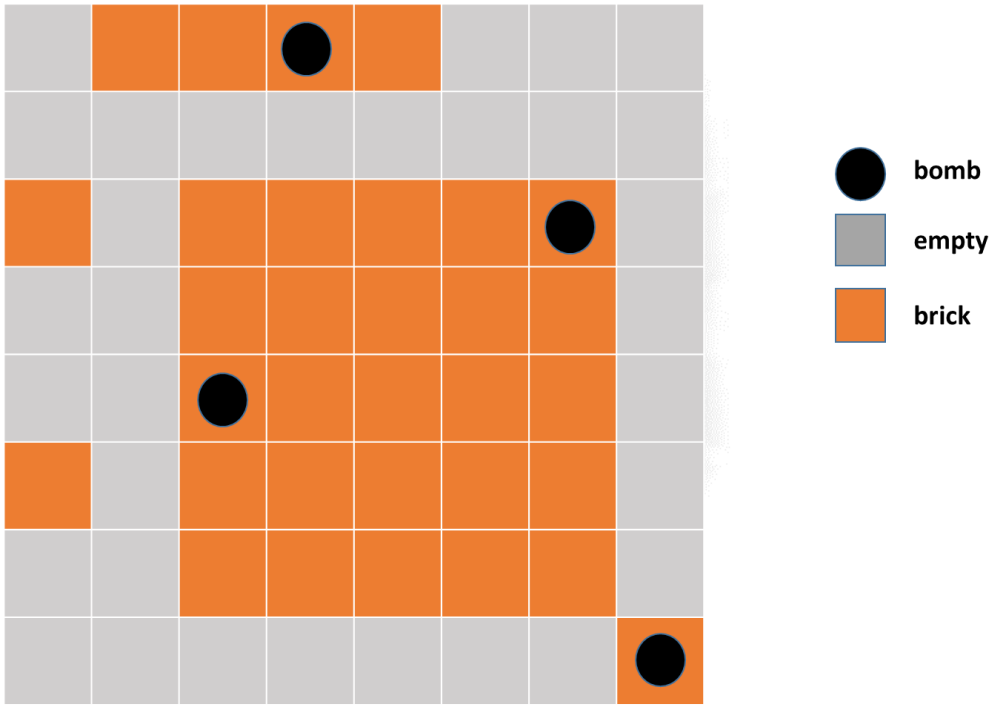
Data Preparation

1. Extract LAB data from TA's directory.
   % tar xvf ~2016dlabta02/OT11_27.tar

Design Description and Examples

Design this simple game of destroying bricks:

There is an 8x8 map.



Input：

You will get 8 inputs numbers from **in [7:0]** and **bomb [7:0]** respectively with **in_valid1** high.

For **in [7:0]**, each bit of inputs: 1 means brick, and 0 means empty.

Place 8 inputs(total bits of input are 64bits) into map as following rule:

input1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
input8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

For **bomb [7:0]**, each bits of inputs: 1 means bomb, and 0 means no bomb.

Bomb positions are same as above.

You will get 10 inputs number from **hit [5:0]** with **in_valid2** high**.**

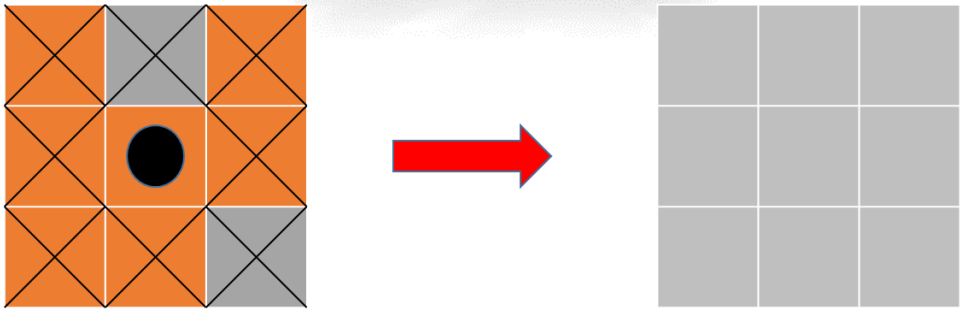
For **hit [5:0]**, each of input means position of brick you need to destroy sequentially.

You are going to come across 3 case when playing this game:

1. hit empty position: nothing happens and get 0 point

2. hit a brick: destroy brick and get 1 point

3. hit a bomb: bomb will destroy the block near of them and itself as following graph:



Notice for bomb: 1. Bombs can destroy other bombs but not behave chain effect on bombs.

2. bombs are bored on positions where bricks are.

3. bombs don’t destroy across boundary.

After finish hitting bricks in this round, output the number of bricks destroyed.

hit number:

7	6	5	4	3	2	1	0
15	14	13	12	11	10	9	8
23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24
39	38	37	36	35	34	33	32
47	46	45	44	43	42	41	40
55	54	53	52	51	50	49	48
63	62	61	60	59	58	57	56

Output：

Ex:

Destroy 3 bricks in this round.

You should output 3 in 1 cycle with out_valid high.

Your goal is to compute these operations by above rules and output the correct answer.

Inputs

1. input data for **in [7:0] and bomb[7:0]** is valid with **in_ valid1** high.
2. **hit [5:0]** is valid with **in_valid2** high.

Input Signals	**Bit Width**	**Description**
clk	1	clock
rst_n	1	asynchronous active-low reset
in	8	Position of brick: 1:brick, 0: empty
bomb	8	Position of bomb: 1: bomb, 0: no bomb
hit	6	Hit position of brick: 0 ~ 63
in_valid1	1	high when **in [7:0], bomb [7:0]** is valid
in_valid2	1	High when hit[5:0] is valid

3. All inputs will be changed at clock ***negative*** edge.

Outputs

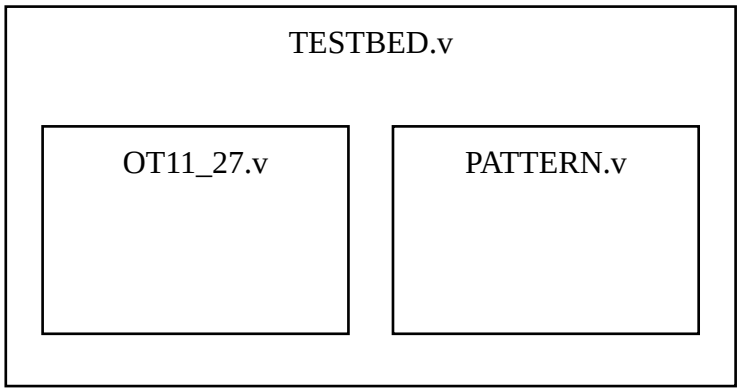
1. Your answer should output at **out [6:0]** with out_valid high.
2. **out_valid** should be low and **out[6:0]** should be set to zero after initial reset.
3. **out_valid** should be set to high when output value is valid.
4. The latency of your design in each pattern should not be larger than 100 cycles.
5. All outputs are synchronized at clock ***positive*** edge.
6. Test pattern will check whether your answer is correct or not at clock **negative edge** when **out_valid** is high.

Output Signals	Bit Width	Description
out	7	output result
out_valid	1	high when **out[6:0]** isvalid.

Specifications

1. Top module name : **OT11_27** (File name : **OT11_27.v**)
2. Input pins: **clk , rst_n , in_valid , in [7:0], bomb[7:0], hit[5:0]**.
3. Output pins: **out_valid , out [6:0]**.
4. **out_valid** should not be raised when **in_valid** is high.
5. It is **active-low asynchronous** reset.
6. The latency of your design in each pattern should not be larger than 100 cycles.

Block Diagram



Note

1. Simulation step:
   - Put your design in 01_RTL
   - Simulation to check design : ./01_run
   - Show wave to debug: nWave &
   - Go to folder 02_SYN/ and check synthesis：./01_run_dc
   - Go to folder 03_GATE/ and check s：./01_run.f
   - Clear up : ./09_clean_up
2. Please add your student ID and name to the file name of .v file before upload file on e3 platform:
   OT11_27_0556123_陳小明.v
3. Sample waveform: