

Sample Problems

1. What does the signal for a static-zero hazard look like?

The signal will always be logic zero except when the hazard occurs which will cause it to temporarily go to logic one (i.e. glitch rises).

2. What causes a static-one hazard?

Two parallel paths for a signal in which (1) one of the signal paths is inverted (2) the two signal paths reconverge into an OR gate.

3. Can a dynamic hazard have an embedded static hazard? When?

Yes. A dynamic hazard always has an embedded static hazard.

4. Can you eliminate a hazard by adding delays to your circuit?

If the delays through the paths of the hazard are exactly matched you can eliminate the hazard. In the real world this is almost impossible to do as the delays will vary with temperature and supply voltage and will be impossible to match, even if you have very fine control of the delays such as you would with a custom chip design.

5. Using a Karnaugh map, how can we tell if a circuit has a hazard?

On a karnaugh map, adjacent but non-overlapping circles are hazards.

6. Using a sum of products Karnaugh map, how can we mask a static-1 hazard?

Find adjacent but non-overlapping circles, form a new circle (using an AND gate) that covers (stays high across) the transition between the original circles.

7. A product of sums karnaugh map can be used to mask what type of hazards. How does the mask get physically implemented?

The product of sums map can be used to mask static-0 hazards. The mask gets physically implemented as an OR gate.

8. What types of hazards are the following?

$x\bar{x}$	static-0 hazard
$x + x\bar{x}$	dynamic hazard

9. When using boolean algebra with hazards what is an important rule to follow?

(1) treat x and !x as separate variable (2) avoid the distributive law (factoring).

10. For the following boolean expression are there any hazards? Find a mask for any hazards.

$$f = (a + b)c + \bar{c}ab$$

$$a=1, b=1$$

$$(1+1)c + \bar{c} \cdot 1 \cdot 1$$

$$c + \bar{c}$$

Yes there is a static-1 hazard. Using a sum of products representation, the mask for the hazard would be $a \cdot b$ to give the final expression.

$$f = (a + b)c + \bar{c}ab + ab$$

11. If you have a boolean expression, what is the first test to see if there is a potential hazard in a given variable.

The first test is to see if the variable and its complement are present in the expression. If not, there is no possibility of that variable causing a hazard.

12. It is always possible to determine if hazards exist by going through all possible combinations of the inputs to a boolean expression. Fortunately, in most cases the analysis can be simplified by using a little thought. Explain how you could simplify the analysis of the following expression.

$$f = (a + b + c)(d + e) + \bar{c}de + (a + b + e)d$$

The only variable that can give a possible hazard is c (i.e. there are both c and !c in the expression). Since c and !c only occur once, for them not to be eliminated d and e have to be 1 (otherwise there would be no !c) and a and b have to be 0 (otherwise there would be no c). Once you set the values of a,b,d and e the analysis is greatly simplified.

13. How do you mask a dynamic hazard?

Embedded in any dynamic hazard is a static-0 or static-1 hazard. Determine the part of the boolean expression that contains the static hazard. Mask the static hazard using regular sum of products or product of sums analysis. Form final expression by combining masked static hazard with remainder (part that doesn't contain the static hazard) of original expression.

14. Can a sum of products circuit have a static-0 hazard?

Only in the careless case where there is a product term that includes a signal and its complement (a useless term that is always equal to 0).

15. Using a Karnaugh map analysis, explain why some two variable changes not maskable?

When the inputs change you travel from one location on the Karnaugh map to another. If both the starting and final location have the same value and you travel through an intermediate location of the opposite value you can generate a glitch (i.e. there is a hazard). You cannot mask this because you cannot overlay the intermediate location with the opposite value, otherwise the overall logic would be incorrect.

16. For multiple variable changes, what are the two possible conditions required for hazard free operation?

- a) All possible Karnaugh map square that may be travelled through have the same value.
- b) Fixing the stationary variable cannot reduce the function to any of the hazard combination (e.g. $a + !a$, $a!a$, etc.)

17. Why don't hazards hurt synchronous circuits?

In synchronous circuits flip-flops only respond on a clock edge. Glitches due to variable changes occur shortly after the clock edge. If the clock period is long enough to allow the glitches to die out they will have no effect.

18. Give an example of a circuit that is sensitive hazards (i.e. glitches).

Any asynchronous circuits (e.g. memories, bus drivers, etc.).

19. What effect do glitches have on the power consumption of a circuit. Why?

They increase the power consumption. In particular, the dynamic power consumption is dependent on the number of nodes charged or discharged in a given period of time. Glitches will increase the number of nodes that are changing state and therefore being charged or discharged.

Second Set

1. What defines the region of a d-flip-flop where the data must hold still?

The region is defined by the setup time and hold time of the d-flip-flop.

2. What do we know about the hold time of most modern flip-flops?

It is zero or negative.

3. Give an example of typical setup and hold times for modern d flip-flop.

Setup time = 2ns , hold time = 0ns

4. What happens if the d input of a flip-flop changes in the restricted region?

The flip-flop output can go 0,1 or metastable.

5. What is the definition of an asynchronous signal?

An asynchronous signal is one that can change in the restricted region around a clock edge.

6. Why is the output of a flip-flop synchronous? Is this always true?

The output is synchronous because the finite output delay guarantees that it will not change in the restricted region of any subsequent flip-flop. This is true as long as the output delay is greater than the hold time and the flip flops clocks change at the same time (i.e. must take into account clock skew).

7. If we have an asynchronous signal being fed into a synchronous circuit, what is the easiest way to make it acceptable?

The simplest way is to pass it through a synchronous flip-flop whose output will be synchronous signal.

8. What is the potential problem with feeding an asynchronous input into a synchronous state machine?

If the input is fed into two or more flip-flops of the state machine it may cause a race condition if its change is captured by one flip-flop and not by another.

9. If we have a synchronous state machine with two unencoded asynchronous inputs what do we know about the state assignment?

Children states must be adjacent on the karnaugh map (i.e. only one stage variable change apart)

10. What is the potential difficulty of flip-flops with asynchronous reset?

In a synchronous state machine all the flip-flops will have a common asynchronous input (reset) which could cause problems when coming out of reset.

11. What is the potential difficulty with a group of asynchronous inputs which are logically related (i.e. encoded)?

We have to ensure that they all get captured at the same time otherwise their encoded values will have no meaning. If there not related to each other we don't care if there captured one or two clock cycles early or late.

12. How can we use coding to minimize problems with encoded asynchronous inputs?

We can code them such that there is only one bit change at any given time (e.g. gray code, thermometer code).

13. What is the basis for using handshaking for multiple encoded asynchronous inputs?

We must know the timing relationship between all the asynchronous inputs so that the synchronous circuit that accepts the handshaking signal will wait for all the inputs to stabilize before it accepts the inputs.

14. How can we use a debouncing circuit to accept a group of encoded asynchronous inputs? Will it always work?

If we sample the inputs two or more times and they do not change then we assume they have stabilized and accept their value. It will work as long as the inputs once changed will stay stable for several clock cycles.

15. Which is preferred, hardware or software debouncing?

If it is fast enough and does not create too much overhead for the processor, software debouncing is preferred because it requires no extra hardware.

16. If you have two synchronous flip-flops with some gates in between what is the minimum clock period?

$$T_{min} = T_{chq} + T_{pd} + T_{setup}$$

17. Why is there a minimum gate propagation delay between two synchronous flip-flops?

If there is clock skew or a large hold time you have to observe a minimum propagation delay or you will have cycle skipping (i.e. signal passing through two flip-flops in one clock cycle).

18. How does negative clock skew affect the minimum and maximum propagation delays between synchronous flip-flops?

It decreases both the minimum and maximum propagation delay. Decreasing the minimum is good (less likely to have cycle skipping) whereas decreasing the maximum is bad (reduces the maximum clock frequency).

19. If a shift register has $T_{pd} = 0\text{ns}$, $T_{chqv} = 1\text{ns}$, $T_{setup} = 1\text{ns}$, $T_{hold} = 0\text{ns}$, $T_{clock} = 10\text{ns}$ and a worst case positive clock skew $T_{skew} = 7\text{ns}$. Will the circuit work correctly?

No. Cycle skipping will occur if $T_{chqv} + T_{pd} < T_{hold} + T_{skew}$. Normally want to keep $T_{skew} < T_{chqv}$

20. In a shift register, what is the advantage of routing the clock against the data shifting direction?

This gives a negative skew which means that we normally won't have to worry about minimum prop delay (i.e cycle skipping) problems.

21. What are the problems with gating a clock signal?

Added clock skew, false clock edges, difficult to test (scan based testing)

22. What is the advantage of gating the clock?

It saves power by turning off the high frequency clock signals to flip-flops.

23. For a positive edge triggered flip-flop that is gated with an AND gate. How do we prevent false clocks?

Make sure there are no positive edges on the enable signal when the clock is high.

24. What is the advantage of gating a positive edge triggered clock with an OR gate instead of an AND gate?

It is easier to design because we only have to worry about its state in the second half of the clock cycle (when clock is low). This means we don't have to worry about minimum prop delays or fast glitches.

25. What is the advantage of adding a transparent low latch to your AND gated positive edge triggered clock.

Like the OR gate it eliminates the problems of minimum delay and fast glitches, unlike the OR gate it gives a full clock cycle (instead of a half clock cycle) for maximum prop delay.

26. What is the advantage of using a clock divider compared to clock gating?

A clock divider uses less power with lower speed clocks during normal operation and when the clocks are turned off. Even when the flip-flop clocks are turned off, a gated clock dissipates power in the high speed clock signal distributed around the chip (i.e. the gating is done locally)