

ASYNCHRONOUS FPGA RISKS

KEN ERICKSON

**California Institute of Technology
Jet Propulsion Laboratory
Pasadena, CA 91109
email: kerickson@jpl.nasa.gov**

*2000 MALPD International Conference
September 26-28, 2000*

AGENDA

- Synchronous versus asynchronous logic designs
- Dealing with asynchronous input signals
- Why marginal designs go undetected
- Asynchronous designs
 - *When to use*
 - *Risks*
 - *Risk Assessment Methods*
 - *Risk Mitigation Methods*
- Examples of Space Hardware FPGA Problems
- Recommendations
- Summary

Synchronous versus Asynchronous Logic

- Synchronous logic

- All flip-flops are clocked using a single global low-skew clock
- Asynchronous inputs are properly synchronized to clock
- FPGA design tools can be used to easily verify that setup and hold time margin exists under worst case conditions for flip-flops

- Asynchronous logic examples

- All flip-flops are not clocked using a single global low-skew clock
- A flip-flop clocked by the output of a gate or another flip-flop
- Asynchronous set and reset inputs of a flip-flop are used for purposes other than initialization
- Multiple clocks are used

Synchronous versus Asynchronous Logic (Cont')

- Asynchronous inputs are not properly synchronized to clock
- Worst Timing analyses
 - ◆ Requires separate timing analyses for each flip-flop
 - ◆ May require many analyses to analyze all possible clock and data paths for a single flip-flop
 - ◆ For a complex design, complete worst case analysis may not be feasible
- FPGA design tools are tailored for easily performing timing analyses of synchronous logic, not complex asynchronous logic
- An asynchronous design may be justified in special instances such as when speed and power requirements cannot be met using synchronous logic

Dealing with Asynchronous Inputs

- When an asynchronous input is clocked into a flip-flop, set up and hold time requirements may not be met
- There is a possibility that the flip-flop can enter into the metastable state
 - The output state is indeterminate for an unpredictable period of time
 - The mean time between metastability events is given by the following equation:

$$\text{MTBF} = \frac{1}{f_{\text{clk}} \cdot f_{\text{dat}} \cdot C1 \cdot e^{-C2 \cdot t_{\text{res}}}}$$

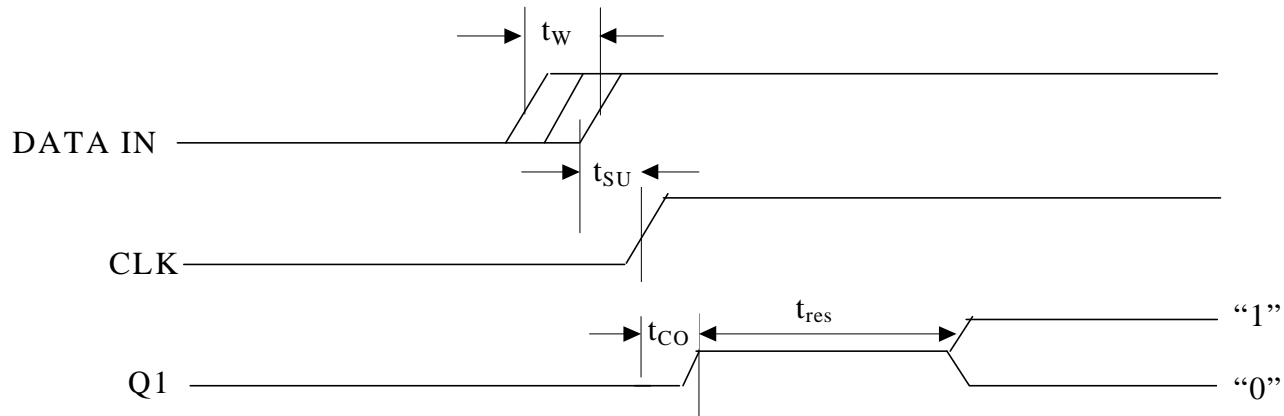
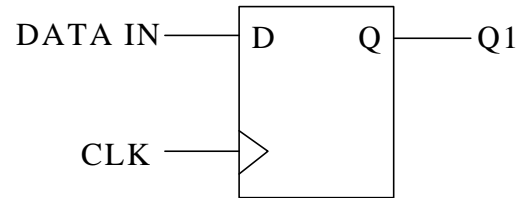
- where:
- f_{clk} = System clock frequency (Hz)
 - f_{dat} = Incoming data rate (Hz)
 - e = Natural log base
 - t_{res} = Resolution time (ns)
 - $C1$ = Describes the likelihood of metastability (device dependent)
 - $C2$ = Describes metastability resolution speed (device dependent)

Dealing with Asynchronous Inputs (Cont')

- Possible consequences

- If the asynchronous input is clocked into multiple flip-flops, the flip flop output states may not all be the same
- If asynchronous parallel data is clocked into a register, the contents of the register may be incorrect
- Functional problems may occur

Metastability



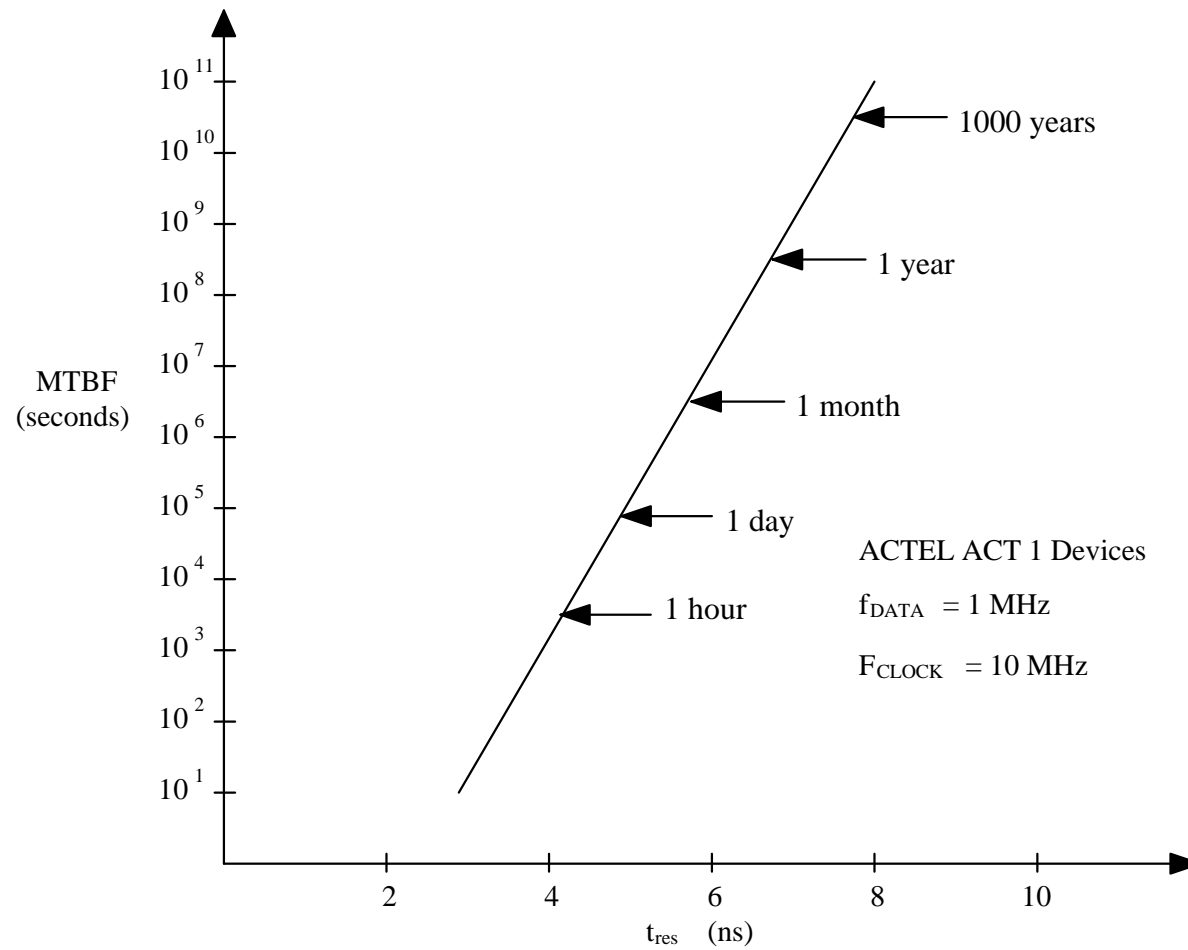
t_w = Time window where input transition may cause a metastable condition

t_{su} = Actual clock setup time for flip-flop

t_{co} = Actual flip-flop propagation delay

t_{res} = Metastability resolution time

MTBF versus Metastability Resolution Time



Dealing with Asynchronous Inputs

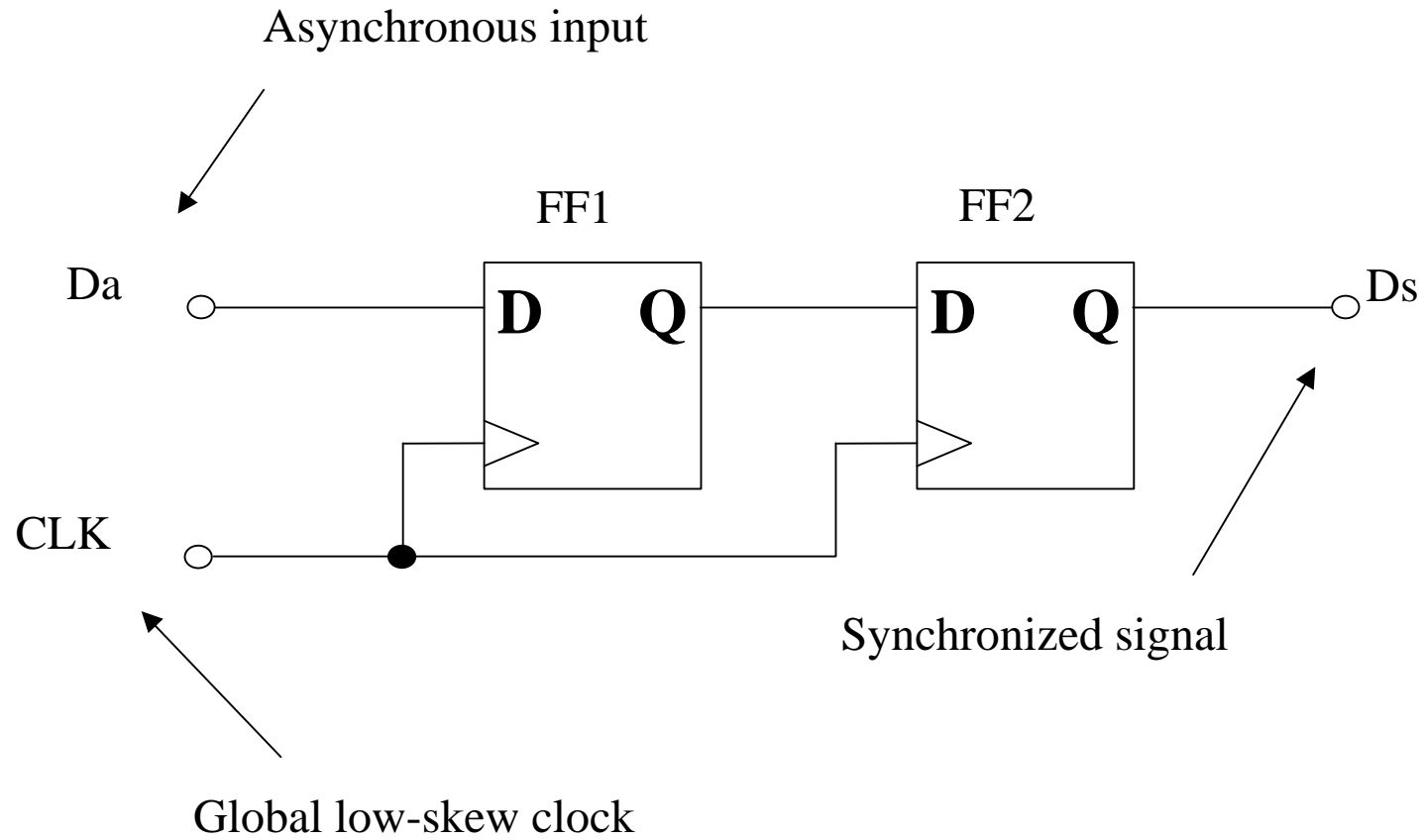
- Synchronizer

- Two cascaded flip-flops clocked by the logic circuit clock
- As long as the metastable state time duration of flip-flop F1 is less than one clock period, a stable logic state is clocked into flip-flop FF2

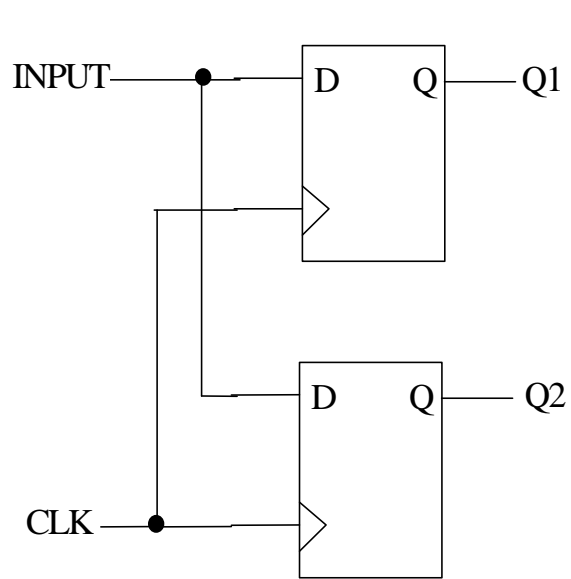
- Design considerations

- Determine the affects of metastability on logic designs
- Attempt to design logic in such a way that the metastability conditions will, at most, result in a self-clearing intermittent condition with no serious consequences
- When the interface signal and the local clock are both derived from the same system clock, determine the worst case timing relationship between the interface signal and the clock to ensure that setup and hold time requirements are met at the interface

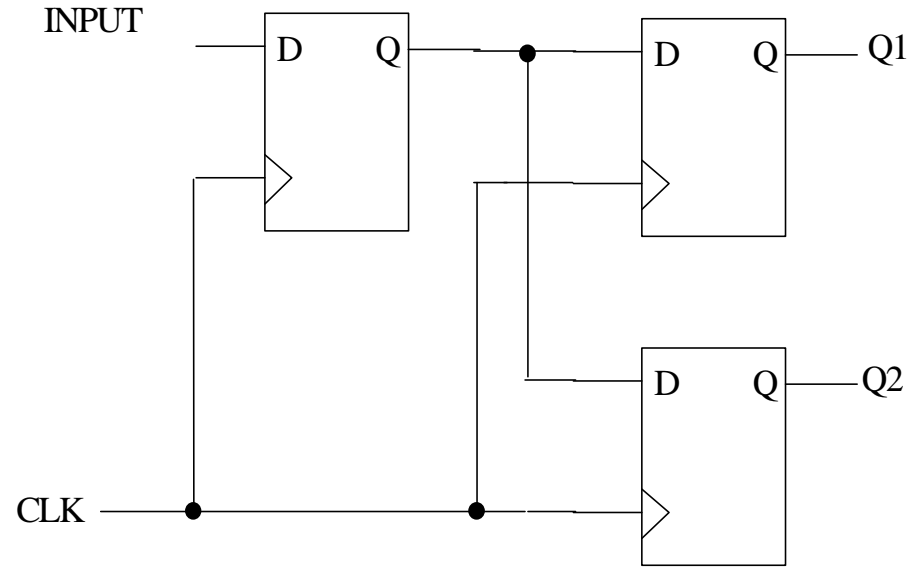
Synchronizer



Synchronizing an Asynchronous Input



Improperly Synchronized



Properly Synchronized

Why Marginal Designs Go Undetected

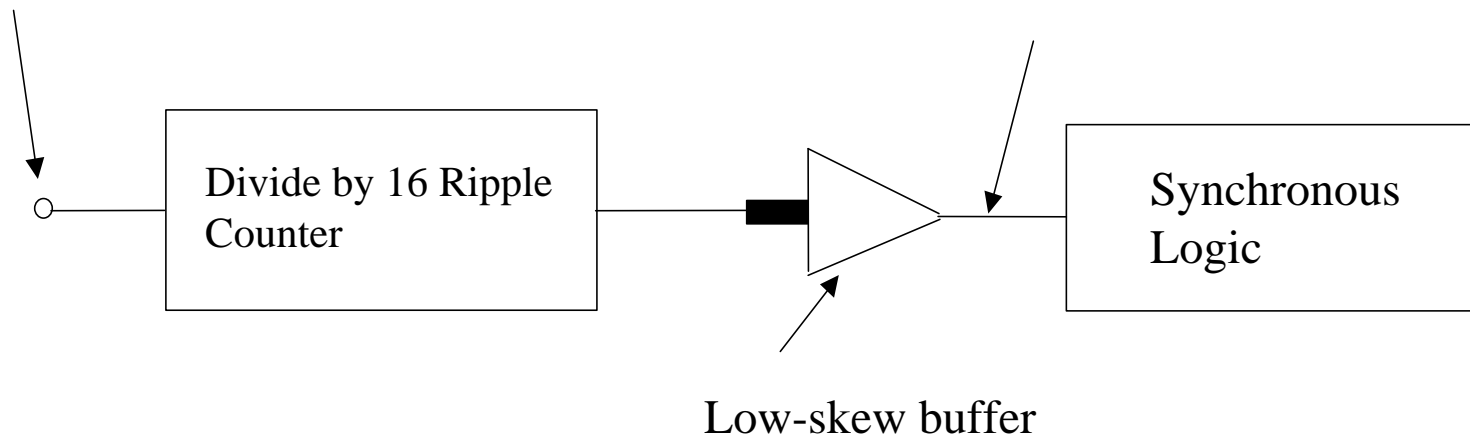
- A detailed peer review of the FPGA design is not performed
 - FPGA-based designs are approved as the result of critical design reviews
 - In some cases the FPGA designs are not completed when the critical design review is held
- Complete worst case timing analysis is not performed
- The risks of asynchronous designs are not assessed and mitigated

When to Use Asynchronous Logic

- When dictated by requirements
 - Asynchronous bus protocol, such as VME, for example
- May be the only way to meet speed requirement
- May be the only way to meet power requirement
- To solve special problems
 - For example, to implement a frequency divider when a low-skew buffer is not available

Frequency Divider

16 MHz high skew clock



Asynchronous Design Risks

- Design may be marginal
- Design fail during the mission due to aging and radiation affects
- Impact of failure may have a significant affect on mission or may result in the end of the mission
- Failures may occur late in the test program, resulting in expensive project delays
- May cause intermittent performance anomalies
- Cause of problems may be difficult to determine

Risk Assessment Methods

- Detailed peer review of design
 - Review worst case timing analysis, if performed
 - What to look for
 - ◆ Potential race conditions
 - ◆ Potential clock skew problems
 - ◆ Gated clocks
 - ◆ Decoder output used as clock or reset input
 - ◆ Potential metastability problems
 - ◆ Asynchronous reset problems
- Complete or partial worst case timing analysis

Risk Assessment Methods (Cont')

- Testing (not preferred method)
 - Use worst case input timing conditions
 - Voltage/temperature margin testing with small temperature steps (such as 1°C)
- Risk assessment considerations
 - Impact of failure on mission
 - Likelihood of failure
 - ◆ This is a judgement call if worst case timing margin is unknown
 - ◆ Consider the mission duration and radiation affects
 - Redundancy does not eliminate risk since both the primary and redundant circuits are susceptible to the same problems

Risk Mitigation Methods

- Redesign logic to make it synchronous and perform timing analysis to verify worst case timing margin – completely mitigates risk
- Perform complete timing analysis
 - May not be feasible if design is complex
 - If complete timing analysis verifies adequate timing margin exist, risk is completely mitigated
- Modify selected portions of design having highest risk – partially mitigates risk
- Perform voltage/temperature margin testing (not preferred method)
 - Use small temperature steps (such as 1°C) since timing problem can occur over a narrow temperature range
 - Risk is only partially mitigated if test is successful
- Prayer - may be needed if risk is not completely mitigated

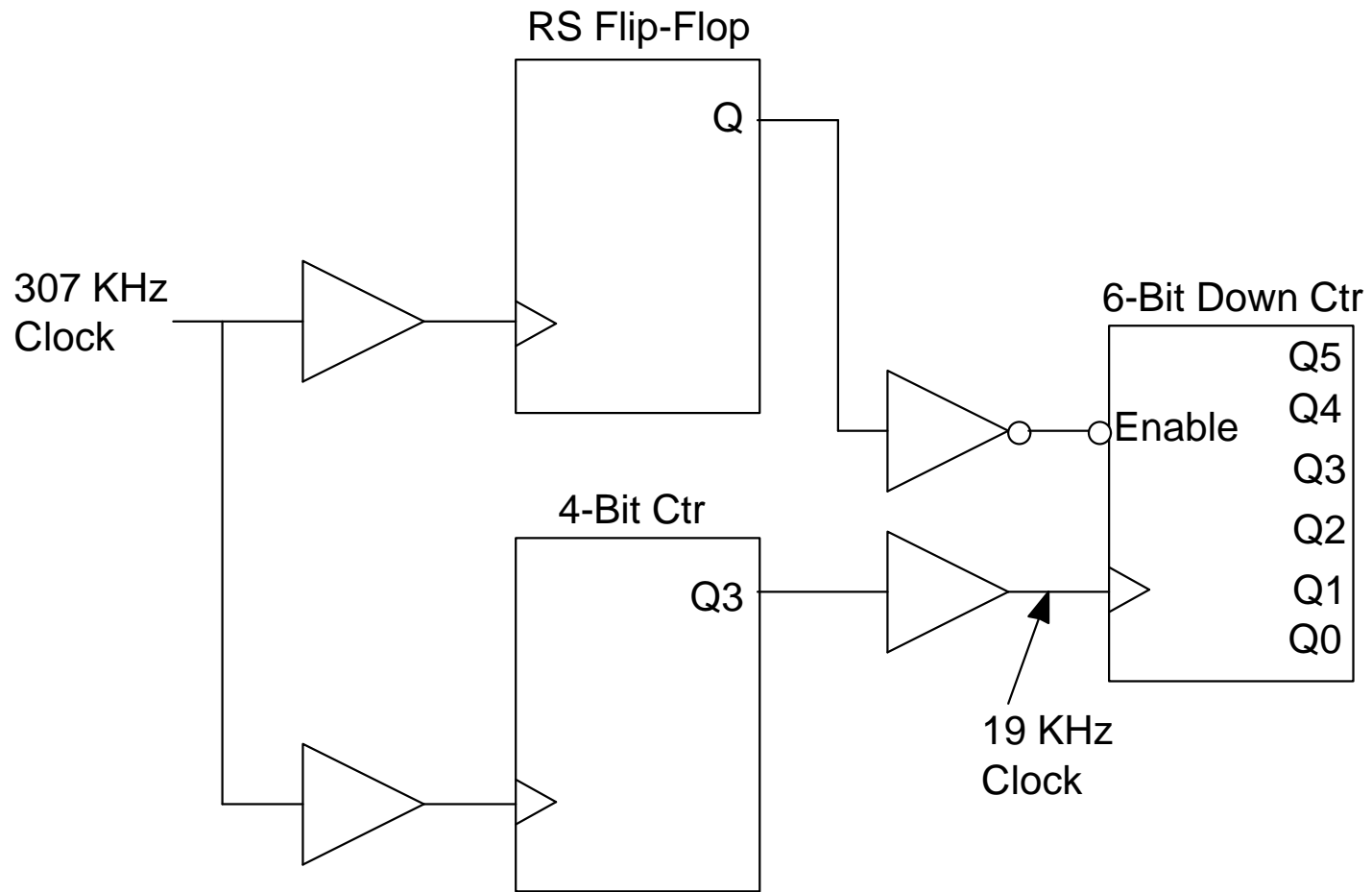
Examples of Space Hardware FPGA Problems

- A spacecraft instrument had many problems during system testing due to marginal asynchronous logic in FPGAs
 - The corrective action was to redesign all FPGAs to make them synchronous
 - Significant additional costs and schedule delays resulted
- A spacecraft experienced inadvertent resets during system testing.
 - The reset problem occurred over the 17 to 20°C temperature range
 - The problem was solved by redesigning FPGAs contained in the flight computer
- Problems were found while performing breadboard testing of a mission critical spacecraft subassembly
 - The following potential problems were found as the result of a peer review of FPGA logic:
 - ◆ The FPGA contained complex asynchronous having unknown timing margin

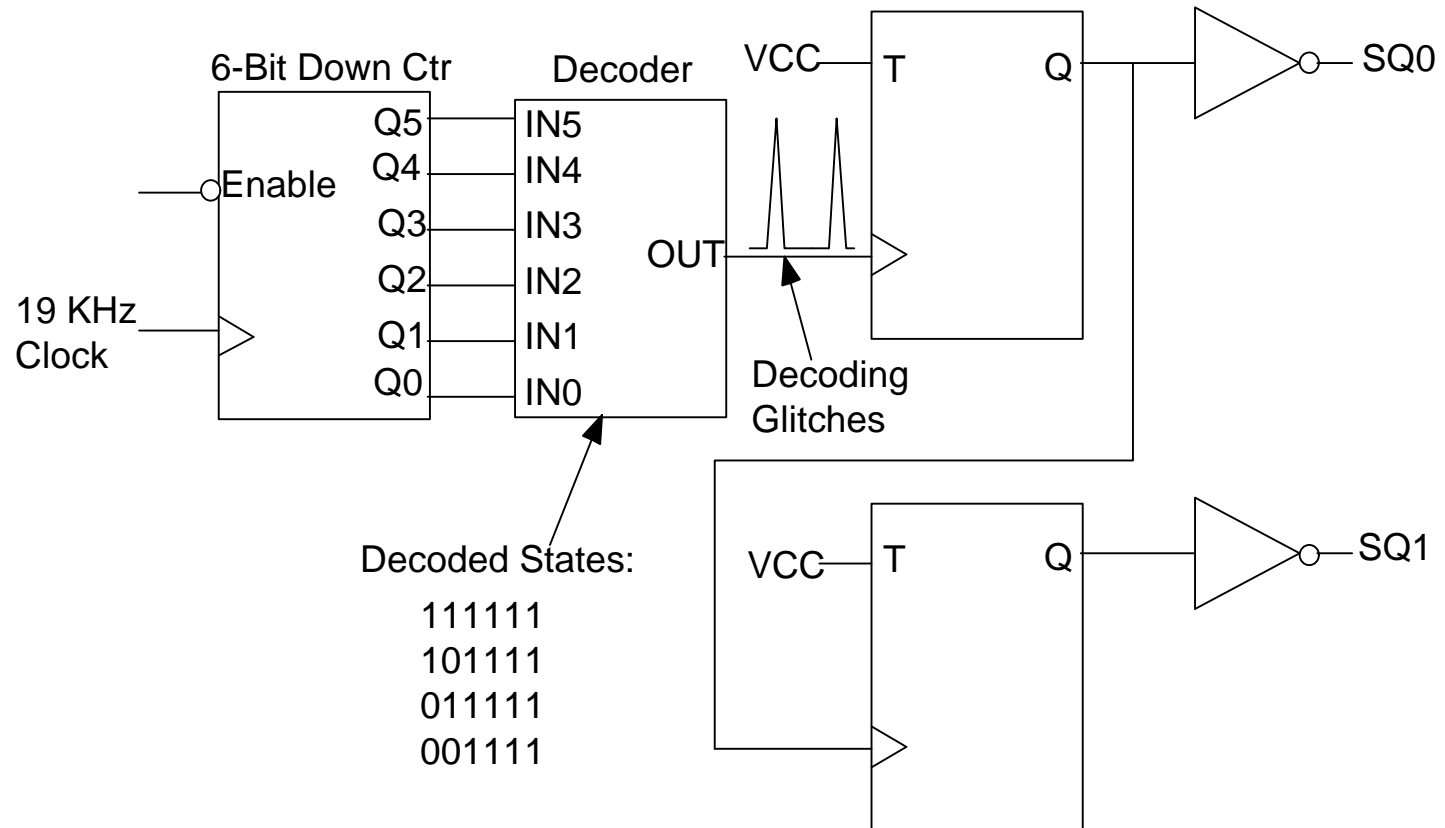
Examples of Space Hardware FPGA Problems (Cont')

- ◆ A potential race condition where the setup time to a flip-flop could not be guaranteed
 - ◆ A decoded counter output was used as a clock (decoding glitches could result in inadvertent clocking)
- The findings of the review along with the following risk assessment were presented to project management:
- ◆ Potential mission ending failure impact
 - ◆ Likelihood of failure is high
- Project management requested that the asynchronous logic be redesigned to make it synchronous
- This redesign was completed and worst case timing margin was verified to completely mitigate the design risk.

Potential Race Condition



Decoder Output Used As Clock



Recommendations

- Prepare a design requirements document for each FPGA-based logic design
 - Ensures that design requirements, including interface signals, are clearly defined
 - Can be reviewed and approved by all affected parties
 - Will help others, including the peer reviewer, understand the design requirements
- Design synchronous logic where possible
- Perform complete worst case timing analyses of all designs
- Conduct a peer review of each logic design by an expert logic designer

Recommendations (Cont')

- Potential problems can be identified
- Even synchronous designs may have potential problems, such as:
 - ◆ Potential problems while powering up FPGA
 - ◆ Critical functions affected by single event upsets
 - ◆ Potential initialization problems
 - ◆ Potential lockup problems
 - ◆ Sneak paths through unpowered logic
 - ◆ Potential timing problems at FPGA to FPGA interfaces
- Assess the risk of all asynchronous logic designs which have not been verified to meet worst case timing requirements and submit to project risk management system
- Mitigate asynchronous logic design risks, if feasible

SUMMARY

- Asynchronous logic designs can result in risks to space missions
- Use asynchronous logic only where required, but perform complete worst case timing analyses
- Assess the risk of all asynchronous logic designs which have not been verified to meet worst case timing requirements
- Mitigate risks where feasible

ACKNOWLEDGEMENTS

- This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.
- Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States government or the Jet Propulsion Laboratory, California Institute of Technology.