Fachhochschule Nordwestschweiz
Hochschule für Technik

ECC 2018

Wie wird ein FPGA-Design sicher für den Einsatz in der Aviatik?

Michael Pichler

Der Auftrag

- Avionics Certifiable Ethernet IP Core
- DO-254
  - Design Assurance Level A

Erste Vorstellungen…

- Komplexer Prozess / Viel Dokumentation
  - Was wird von DO-254 verlangt?
- Der Ethernet Core muss absolut deterministisch sein
  - Nur Punkt-zu-Punkt Verbindungen?
- Die Wahl der richtigen FPGA Technologie ist wichtig
  - Anti-Fuse Technologie von MicroSemi, oder doch
  - SRAM Technologie von Intel/Xilinx?

- **WIR BRAUCHEN ZUERST EINE SCHULUNG!**

…dann die Gewissheit

- Es gibt viel Papierzeugs und störende Strahlungen

1

2

Papierzeugs

Papierzeugs

Radio Technical Commission for Aeronautics (RTCA)

- RTCA sagt nicht, wie es geht
- RTCA definiert, wie man vorgehen soll
  - Planung
  - Design
  - Validation & Verifikation
  - Konfiguration
  - Prozesssicherheit

Certification Authorities Software Team (CAST)

- Position Paper CAST-33
- Vereinfachter Prozess für IPs, die in FPGA eingesetzt werden


Certification Authorities Software Team (CAST)

Position Paper CAST-33

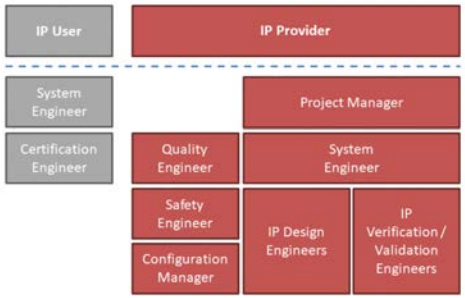
Compliance to RTCA DO-254/ EUROCAE ED-80, "Design Assurance Guidance for Airborne Electronic Hardware", for COTS Intellectual Property Used in Programmable Logic Devices and Application Specific Integrated Circuits

Das Vorgehen planen

- PHAC
  - Plan for the Hardware Aspects of Certification
  - 26-Seiten Dokument
- DAP
  - IP Design Assurance Plan
  - 38-Seiten Dokument

und erst dann beginnen!

Projektorganisation



100% Functional Coverage

Der System Engineer schreibt die Spezifikation und definiert **Requirements**

Das Designteam implementiert **Design Items**

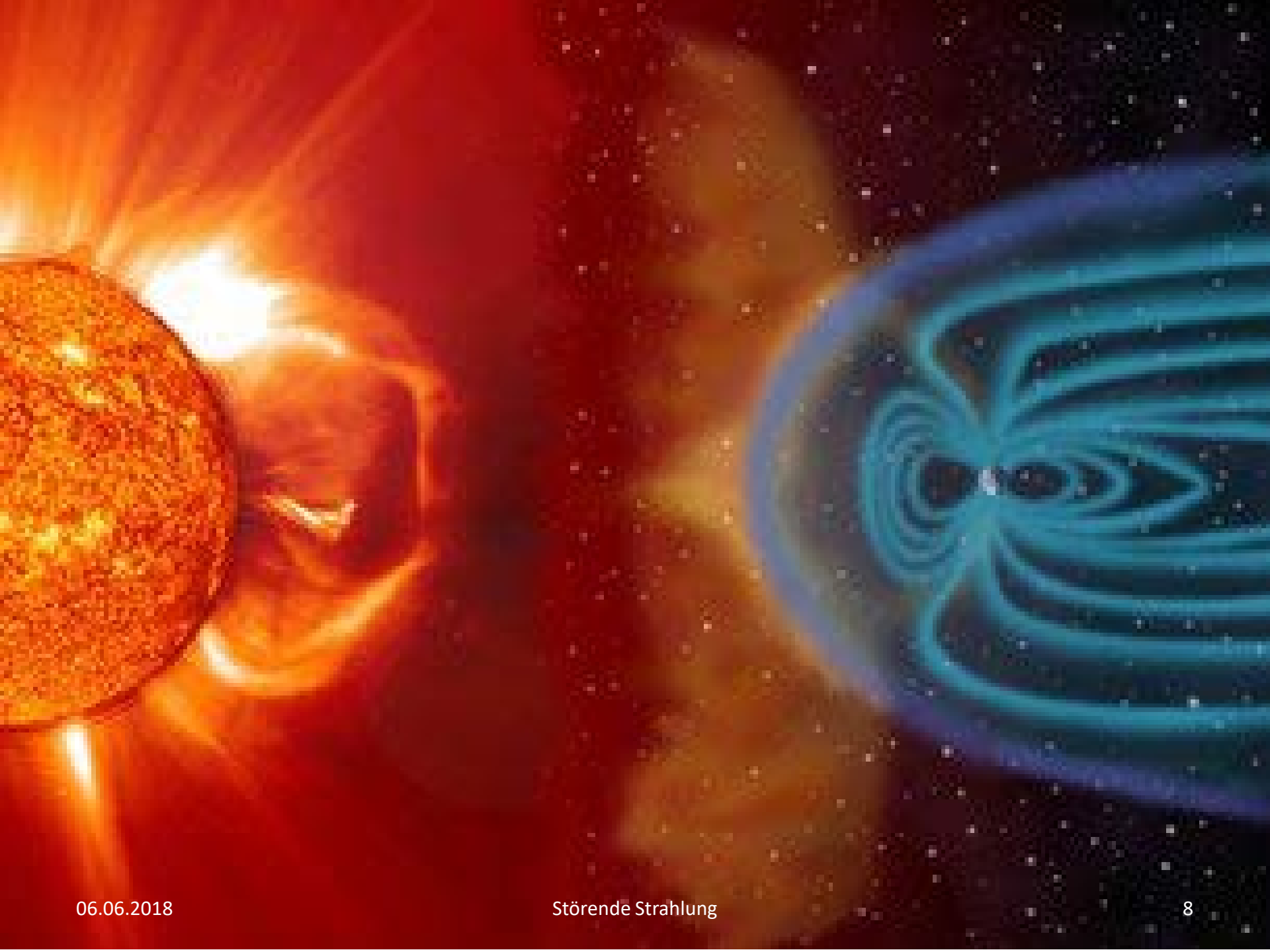
Das Verifikations-team prüft mit **Test Cases**

2 different Simulators

100% Code Coverage

- Automatisch
  - Statements
  - Branches
  - Conditions
  - Expressions
  - Finite State Machines
- Manuell
  - Nicht detektierbare Code-Elemente
  - Redundante Logik
    - z.T. maskiert mit Pragmas

Störende Strahlungen



SEU Mitigation

- Neutronenfluss
  - Hochenergetische Neutronen führen zu SEU (Single Event Upset)
  - Abhängig von der Höhe und dem Breitengrad, Maximum bei
    - 20'000 m.ü.M.
    - 90. Breitengrad
  - Unterschied gegenüber Meereshöhe (New York)
    - Faktor 1400
  - Worst Case Berechnungen vom Safety Engineer
    - Nur ein SEU pro Frame
    - Für ein Bit: 0.72 FIT (Failure in Time)
    - Bei 10'000 Flip-Flops: 7200 FIT

Erlaubte Fehlerraten

Design Assurance Level (DAL)	Description	Target System Failure Rate	Example System
Level A (Catastrophic)	Failure causes crash, deaths	$< 1 \times 10^{-9}$	Flight controls
Level B (Hazardous)	Failure may cause crash, deaths	$< 1 \times 10^{-7}$	Braking systems
Level C (Major)	Failure may cause stress, injuries	$< 1 \times 10^{-5}$	Backup systems
Level D (Minor)	Failure may cause inconvenience	No safety metric	Ground navigation systems
Level E (No Effect)	No safety effect on passeger/crew	No safety metric	Passenger entertainment

Signalanalyse

- Für jedes Flip-Flop im FPGA wird untersucht
  - Lebensdauer innerhalb eines Ethernet Frames [ns]
    - Die daraus resultierende Fehlerrate berechnen
    - Mögliche Folgen ableiten
  - Erforderliche Gegenmassnahme
    - Not Critical
    - Error Detection
    - Error Correction
  - Mögliche Verbesserungsvorschläge
    - Entscheid: Ja/Nein
    - Implementiert: Ja/Nein
- Dies ergibt eine grosse Excel-Tabelle

Signalanalyse

Signal Analysis			
Failure rate for 1 bit	7.20E-10		
Failure rate required	1.00E-09		(DAL-A)
Interval short frames	872	ns	
Interval long frame	15432	ns	
Number of listed signals	298		
Number of analysed signals	298		(100%)
Number of green signals (with error detection or not critical)	263		(88%)
Number of grey signals (no error detection)	23		(8%)
Number of red signals (no error detection and high failure rate)	12		(4%)
Number of listed registers	39898		
Number of analysed registers	39898		(100%)
Accumulated failure rate of unprotected registers	4.590E-08		(red if DAL-A can't be achieved)
Accumulated failure rate of registers without correction	2.366E-05		

i0_ace_unit/i0_udp_core/i0_rx/i0_checksum_gen	payload_cksm	16	872	100.0	15432	100.0	1.15E-08	yes			SEU will lead to a wrong checksum and thus the packet will be discarded		
i0_ace_unit/i0_udp_core/i0_rx/i0_checksum_gen	payload_cksm_ready	1	872	100.0	15432	100.0	7.20E-10	yes			SEU could lead to an early finish of the checksum which results in a wrong checksum and thus the packet will be discarded		
i0_ace_unit/i0_udp_core/i0_rx/i0_checksum_gen	payload_cksm_tmp	16	144	16.5	11760	76.2	8.78E-09	yes			SEU will lead to a wrong checksum and thus the packet will be discarded		
i0_ace_unit/i0_udp_core/i0_rx/i0_checksum_gen	udp_checksum	16	20	2.3	20	0.1	2.64E-10	yes			SEU will lead to a wrong checksum and thus the packet will be discarded		
i0_ace_unit/i0_udp_core/i0_rx/i0_checksum_gen	udp_cksm	16	176	20.2	11792	76.4	8.80E-09	yes			SEU will lead to a wrong checksum and thus the packet will be discarded		
i0_ace_unit/i0_udp_core/i0_rx/i0_checksum_gen	udp_descriptor_valid_d1	1	8	0.9	8	0.1	6.61E-12				SEU will lead to a dead-lock in calculation and causes the rx_mux to miss a frame		No improvements because probability is very low already.
i0_ace_unit/i0_udp_core/i0_rx/i0_checksum_gen	udp_length	11	16	1.8	16	0.1	1.45E-10	yes			SEU will lead to a wrong checksum and thus the packet will be discarded		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_tx/i0_crc32	crc	32	48	5.5	48	0.3	1.27E-09	yes			SEU will lead to a wrong FCS		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_tx/i0_crc32	crc_reg	32	536	61.5	12168	78.8	1.82E-08	yes			SEU will lead to a wrong FCS		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_tx/i1_crc32	crc	32	160	18.3	48	0.3	4.23E-09	yes			SEU will lead to a wrong CRC		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_tx/i1_crc32	crc_reg	32	424	48.6	12168	78.8	1.82E-08	yes			SEU will lead to a wrong CRC		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_rx/i0_crc32	crc	32	8	0.9	8	0.1	2.11E-10	yes			SEU will lead to a wrong FCS		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_rx/i0_crc32	crc_reg	32	528	60.6	12160	78.8	1.82E-08	yes			SEU will lead to a wrong FCS		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_rx/i1_crc32	crc	32	196	22.5	68	0.4	5.18E-09	yes			SEU will lead to a wrong CRC		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_rx/i1_crc32	crc_reg	32	416	47.7	12160	78.8	1.82E-08	yes			SEU will lead to a wrong CRC		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_rx/i0_fcs_checker	data_buf	8	480	55.0	12112	78.5	4.52E-09	yes			SEU will cause an error in the header checksums or the integrity CRC		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_rx/i0_fcs_checker	data_last_buf	1	480	55.0	12112	78.5	5.65E-10	yes			SEU will cut off the frame		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_rx/i0_fcs_checker	data_valid_buf	1	480	55.0	12112	78.5	5.65E-10	yes			SEU will cut off the frame		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_rx/i0_fcs_checker	rx_mac_data	8	480	55.0	12112	78.5	4.52E-09	yes			SEU will cause an error in the header checksums or the integrity CRC		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_rx/i0_fcs_checker	rx_mac_error	1	480	55.0	12112	78.5	5.65E-10	yes			SEU will induce an error which leads to a packet drop		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_rx/i0_fcs_checker	rx_mac_last	1	480	55.0	12112	78.5	5.65E-10	yes			SEU will cut off the frame		
i0_ace_unit/i0_udp_core/i0_mac/i0_mac_rx/i0_fcs_checker	rx_mac_valid	1	480	55.0	12112	78.5	5.65E-10	yes			SEU will cut off the frame		

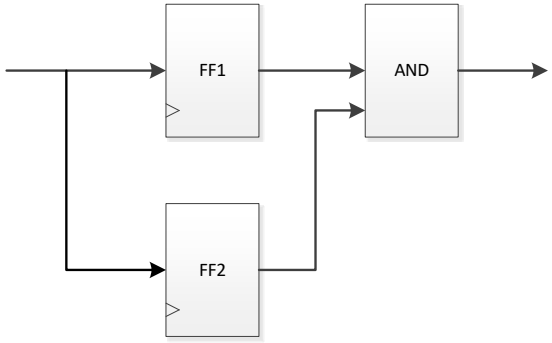
Strategie

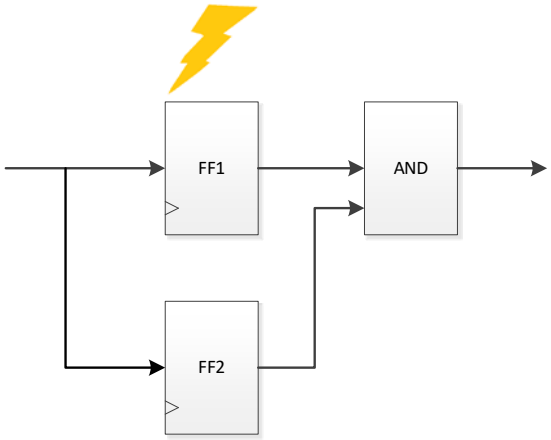
- Im Zweifelsfalle:
  - Lieber ein korrektes Frame als fehlerhaft maskieren
    - indem man die Frame Checksum (FCS) interviert
  - als ein fehlerhaftes Frame als korrekt anerkennen!

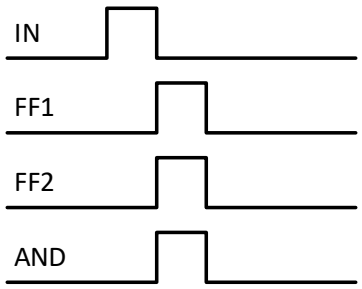
Gegenmassnahmen

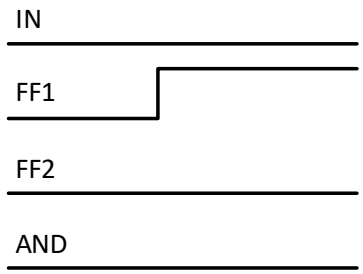
- Datensignale
  - z.Bsp. Kontrollbit
    - DMR (Double Modular Redundancy), 2 Flip-Flops, 1 Entscheider
    - TMR (Triple Modular Redundancy), 3 Flip-Flops, 1 Entscheider
- Datenvektoren
  - z.Bsp. Konfigurationsdaten
    - Hamming Code mit Hamming-Distanz 4
    - 2 Fehler erkennbar, 1 Fehler korrigierbar
- Datenpakete
  - z.Bsp. Ethernet Packet
    - CRC-Checksummen

DMR für Pulse

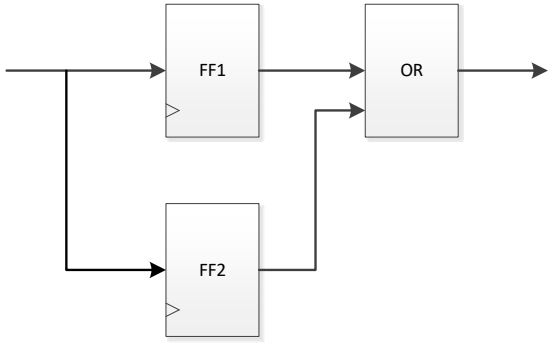


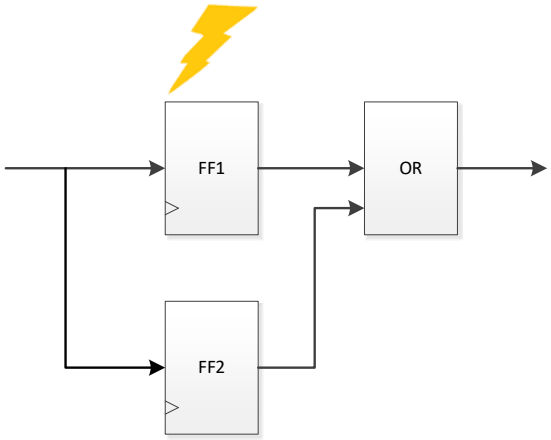


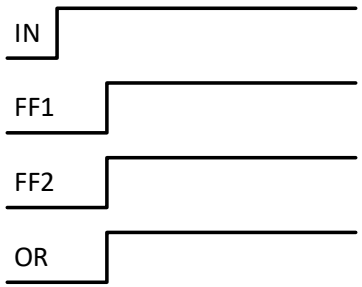


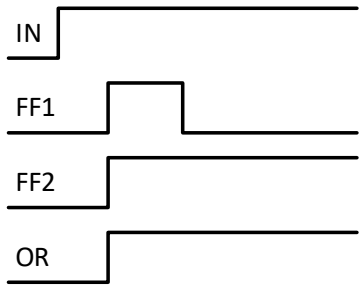


DMR für Error Flags



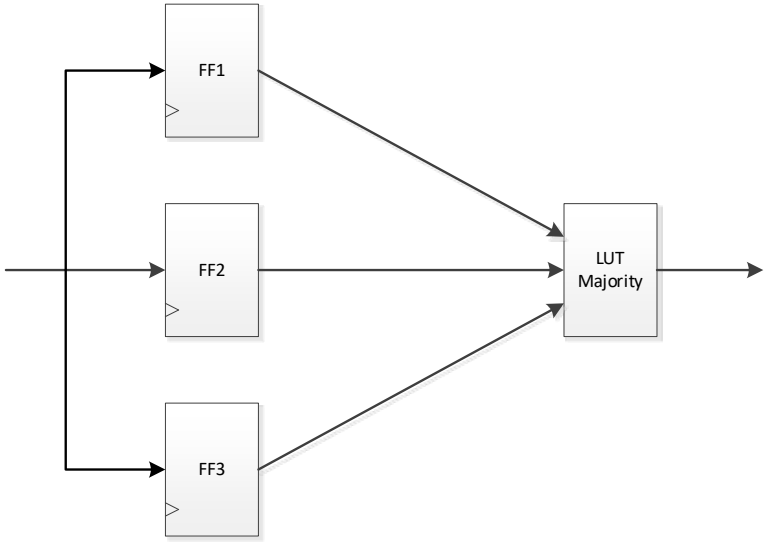






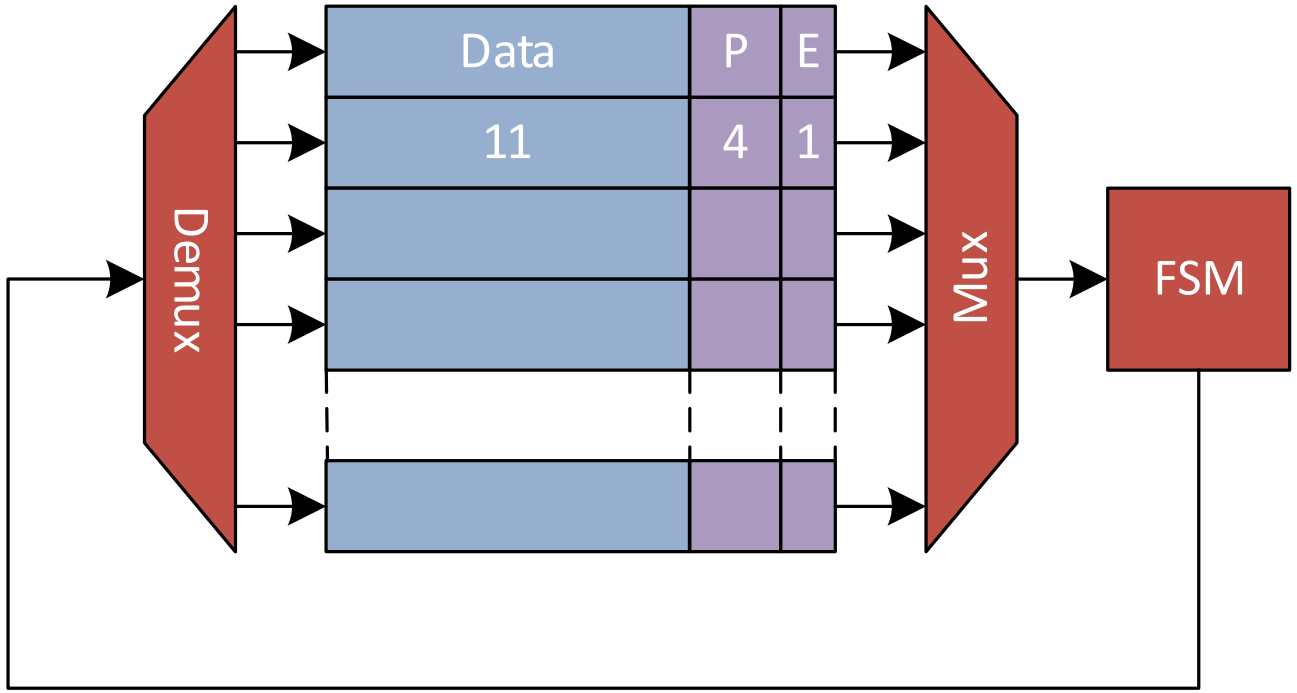
TMR

- Tripple Modular Redundancy = Hamming (3,1)
  - 1 Datenbit
  - 2 Paritybits
- Wir verwenden 3 Flip-Flops und ein Entscheider
  - Mehrheitsentscheider



Hamming

- 39 x 11 Datenbits werden gesichert mit
  - Hamming (16,11)
  - 11 Datenbits, 4 Parity Bits, 1 Extra Parity Bit



Hamming

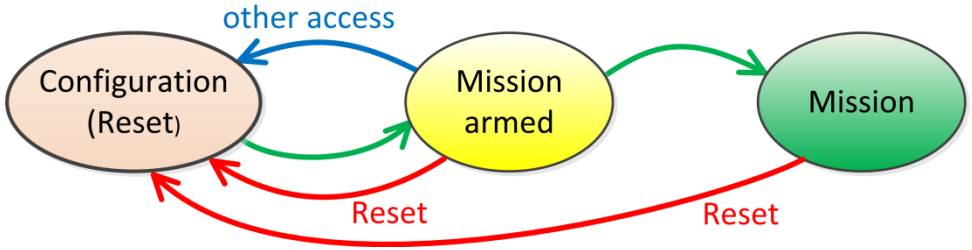
Extra Parity	Syndrom == 0	Syndrom =! 0
0	No Error	Double Error
1	Single Error in Parity Bit	Single Error in Codeword

CRC

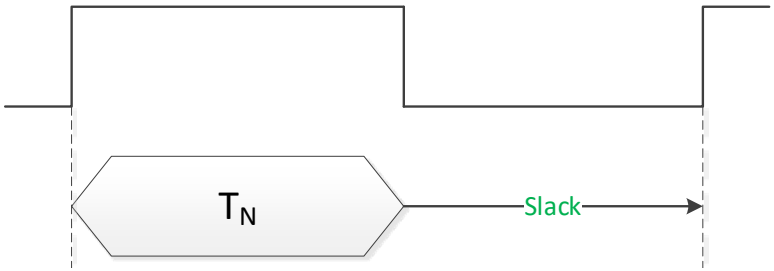
- Für die Checksummen wird ein CRC-32 verwendet
  - Generator Polynom: 0x104C11DB7
    - $X^{32} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + x^0$
- CRC-32 genügt nicht für Jumbo Frames
  - Wir unterstützen Ethernet Pakete bis 1500 Bytes
- Wird verwenden den CRC-32 mehrfach
  - Für die Payload zwischen Host und IP
  - Für das Ethernet Paket zwischen zwei IPs

Zustandsmaschinen

- Generell: Vivado bietet FSM-Codierung mit
  - Hamming 2 (Fehlerdetektion, Default-State)
  - Hamming 3 (Fehlerkorrektur, Auto-Save-State)
- Ausnahme: Master-FSM
  - Die Master-FSM hat drei Zustände und wird mit 6 Flip-Flops codiert:
    - "000000" Configuration-Mode
    - "000111" Mission Armed Mode
    - "111111" Mission Mode



SEU & Timing-Betrachtungen



Normal Case

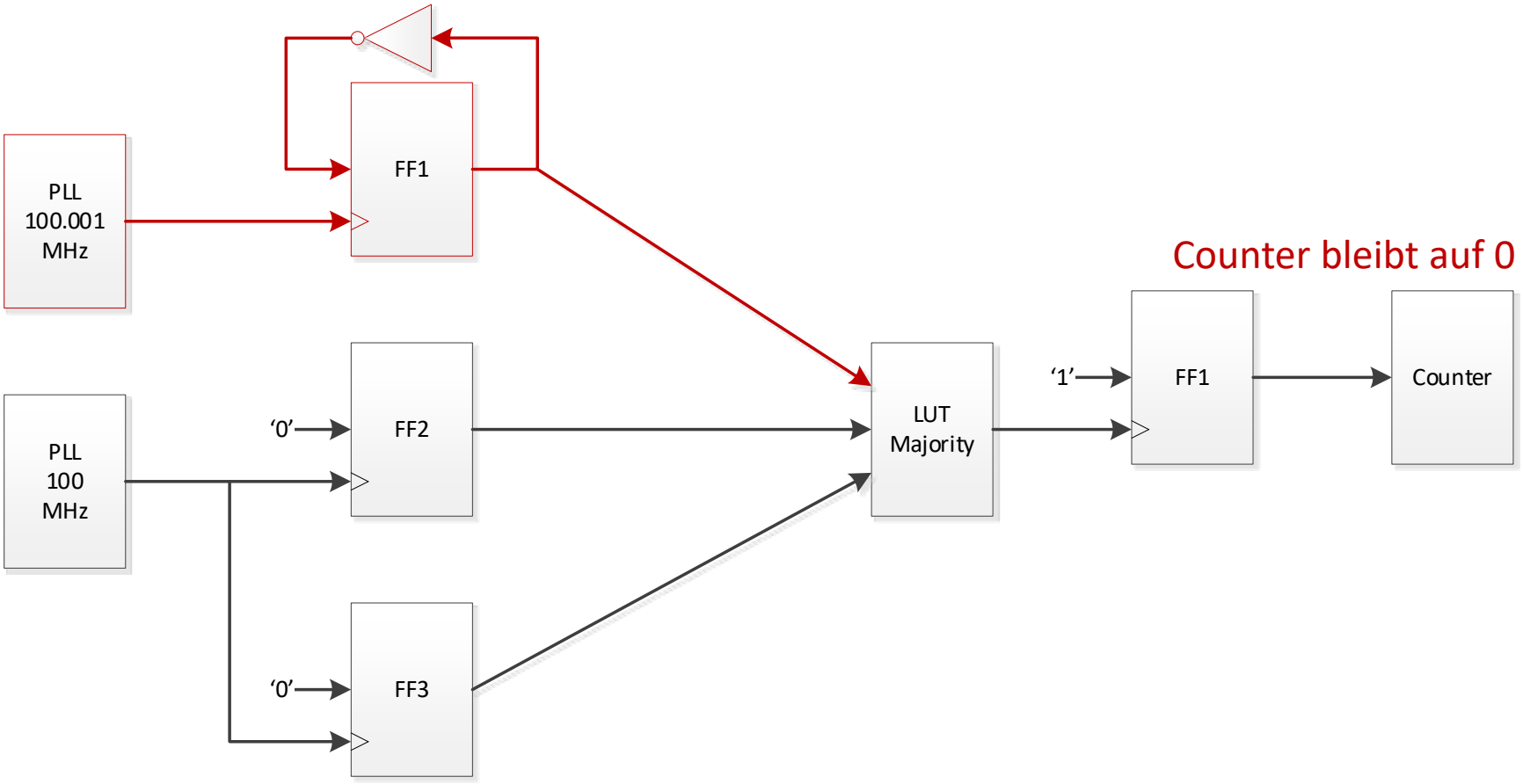
Hypothese

- 1. Ohne Timing-Betrachtungen funktioniert alles tadellos
- 2. Mit Timing-Betrachtungen (meistens) nicht!
  - Ein SEU kann irgendwann auftreten!
  - T_{SEU} ist zwischen 0 ns und T_{clk}
  - Potentielle Fehlerfälle entstehen, wenn $T_{SEU} + T_S >= T_{clk}$
- 3. Die Wahrscheinlichkeit, dass es schief geht
  - $p = 1 - Slack/T_{clk}$

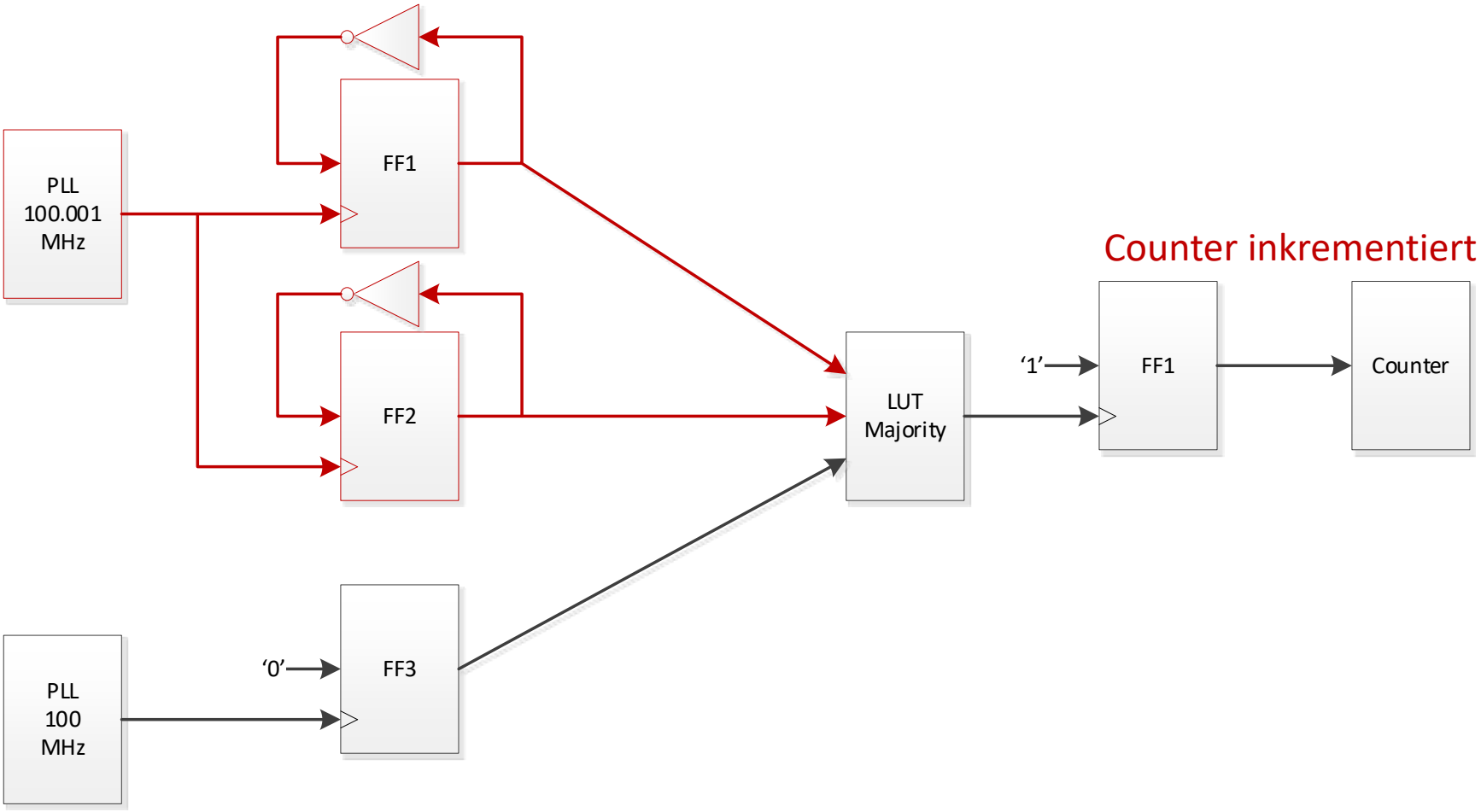
Stimmt diese Hypothese?

- Was wir zu wissen glaubten:
  - Änderungen an den Eingängen von FPGA-LUTs erzeugen Glitches am LUT-Ausgang (bei FPGAs mit SRAM Technologien)
- Wenn dem so ist, funktionieren DMR, TMR und Hamming Codes nicht zuverlässig

TMR: Messung 1



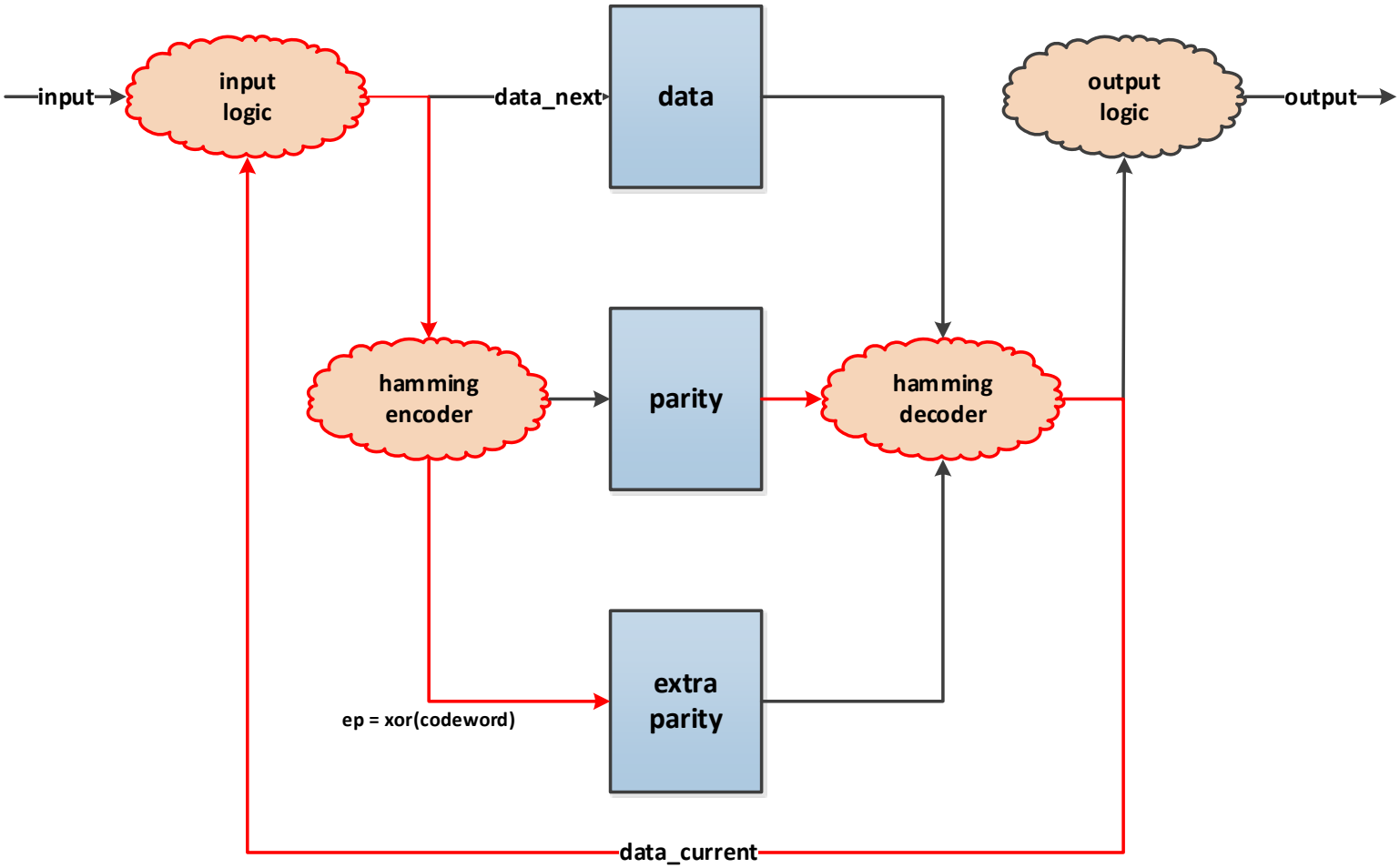
TMR: Messung 2



TMR: Fazit

- TMR
  - Der Entscheider kann mit einer LUT realisiert werden
  - Entscheidung funktioniert zuverlässig, wenn wir annehmen, dass nicht zwei SEU innerhalb von einem Taktzyklus auftreten

Hamming: Timing



Hamming: Fazit

- Die Hamming Coder und Encoder erzeugen mit grosser Wahrscheinlichkeit eine mehrstufige Logik
- Wir verwenden Hamming Codes (16,11) bei den Konfigurationsdaten
  - Glitches sind zu erwarten
  - Wenn das Syndrom oder das Extra Paritiy des Hamming Decoders ungleich Null sind
    - wird das aktuelle Frame als fehlerhaft markiert
    - wird die aktuelle Konfiguration zuerst nicht geändert. Erst nach einer zweiten Dekodierung werden Korrekturen vorgenommen

Extra Parity	Syndrom == 0	Syndrom =! 0
0	No Error	Double Error
1	Single Error in Parity Bit	Single Error in Codeword

Zusammenfassung

Zusammenfassung

- Avionics Certifieable Ethernet IP nach DO-254 (DAL-A)
  - DAL-A
    - Fehlerrate < 10^{-9}
  - SEU
    - Können auftreten → Gegenmassnahmen
  - Nur Fehlererkennung
    - DMR
    - CRC
    - One-Hot FSM mit Hamming 2
  - Mit Fehlerkorrektur
    - TMR
    - Hamming 4

Kontakt

Fachhochschule Nordwestschweiz
Hochschule für Technik
Institut für Mikroelektronik

Michael Pichler
Steinackerstrasse 5
CH-5210 Windisch

+41 56 202 75 26
michael.pichler@fhnw.ch
www.fhnw.ch/ime