

Prüfen G.709-gemäßer FEC-Implementierungen

Sicherstellung korrekter FEC-Funktionalität

Instrumente zum Prüfen von Glasfaser-Netzen müssen in der Lage sein, die Implementierung von FEC-Algorithmen (Forward Error Correction) und die FEC-relevanten Parameter eingehend zu analysieren, damit sie ein umfassendes Funktionsangebot für den FEC-Test bieten. Dieser Artikel wurde mit dem Ziel geschrieben, das Verständnis und die Nutzung der Vorteile, die diese Funktionalität bietet, zu erleichtern. Der Beitrag informiert nicht nur über die in Testlösungen erforderlichen FEC-Manipulationsmöglichkeiten, sondern befasst sich auch mit den implementierten FEC-Testfunktionen, die dem Anwender helfen, den Design- und Verifikations-Zeitaufwand bei der Entwicklung und Einrichtung von FEC-Systemen zu verringern.

FEC-Tests sind unerlässlich, um die korrekte Implementierung des FEC-Algorithmus in FEC-Encodern und -Decodern sowie in F&E-Umgebungen zu verifizieren und um Netzelemente (NE) einer Höchstbeanspruchungsprüfung (Maximum Stress Test) zu unterziehen, in der sie mit der größtmöglichen Anzahl Fehler konfrontiert werden. Notwendig ist dies beispielsweise bei der Durchführung von Systemverifikationen, in denen die korrekte Installation (Einrichtung und Integration) von NEs überprüft wird, sowie bei der Endprüfung von Installationen.

FEC-Implementierung in ITU-T G.709

Die Richtlinie G.709 verwendet den Reed-Solomon-Code RS(255/239). Dieser sieht die Verwendung von 239 Bytes als Informationsbytes vor, aus denen die aus 16 Bytes bestehende FEC-Parität (255-239) errechnet wird. Ein OTN-Frame besteht aus vier OTU-Zeilen

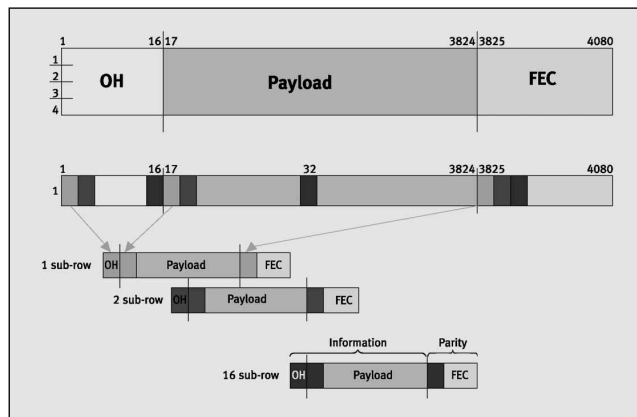


Abb. 1:
Aufbau eines OTN-Frames

(Optical channel Transport Unit). Jede OTU-Zeile gliedert sich wiederum in 16 Unterzeilen (Codeworte) aus jeweils 255 Bytes. Die Unterzeilen werden nach dem Byte-Interleaved-Prinzip gebildet. Das heißt, dass die erste Unterzeile jeweils das erste OH-Byte, das erste Payload-Byte (Nutzdaten) und das erste FEC-Byte enthält. Das FEC-Byte jeder Unterzeile wird in Byteposition 240 eingefügt. Diese Angaben gelten für alle 16 Unterzeilen. Abb. 1 illustriert den Prozess.

Der Reed-Solomon-Code ist in der Lage, pro Unterzeile 16 Byte-(Symbol-)Fehler zu erkennen oder bis zu acht Bytefehler zu korrigieren. Abb. 2 verdeutlicht die verschiedenen FEC-Einführungsmethoden eines bestimmten FEC-Tester-Typs.

FEC-Prüffunktionen

Im Interesse effektiver und präziser FEC-Tests sollte bei der Auswahl einer geeigneten Testlösung darauf geachtet werden, dass die nach-

folgend aufgezählten Funktionen vorhanden sind.

Fehlereinfügung

Mit dieser ‚Error Insertion‘-Funktion ist es dem Installateur möglich, den FEC-Algorithmus auch beim Einsatz im Feld zu verifizieren. Sollte während der Testprozedur ein Ausfall auftreten oder ein Funktionsfehler festgestellt werden, ist eine weitergehende Analyse nötig.

Maximum Stress Test

Derartige Höchstbelastungsprüfungen eignen sich, um einen Prüfling mit der maximalen Fehleranzahl zu beaufschlagen, die er bei Einhaltung der ITU-T-Richtlinie G.709 korrigieren können muss. Ebenso wie bei der Fehlereinfügung gilt auch hier, dass eine gründlichere Untersuchung fällig wird, wenn ein Ausfall oder ein Funktionsfehler festgestellt wird.

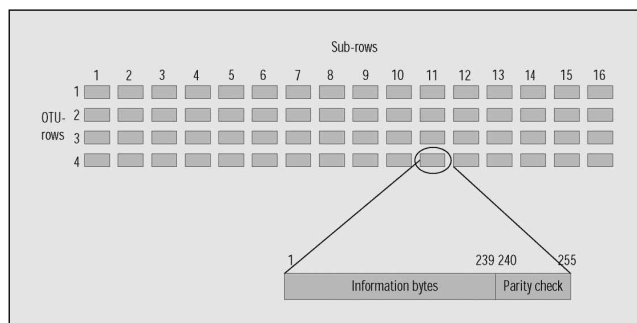


Abb. 2:
Diese Matrix illustriert die vier OTU-Zeilen und sämtliche 16 Unterzeilen

Autor

Dr. ANDREAS KAUFMANN
Acterna Eningen GmbH;
Mühleweg 5, D-72800 Eningen u. A.
Fon: 07121/861884, Fax: 07121/862054
E-Mail andreas.kaufmann@acterna.com

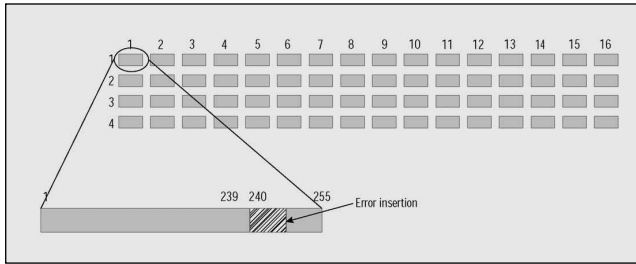


Abb. 3: Einfügen von Fehlern in die erste Unterzeile der ersten OTU-Zeile

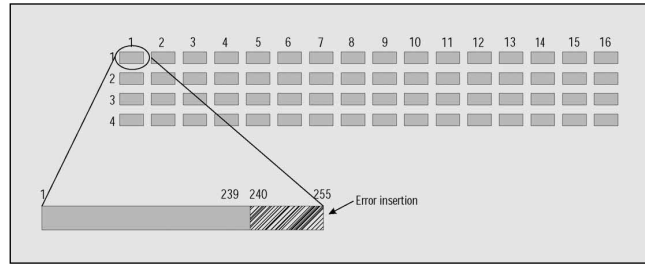


Abb. 4: FEC: Einfügen nicht korrigierbarer Fehler

Anspruchsvollere FEC-Tests

Die Fähigkeit, bestimmte Unterabschnitte sämtlicher möglichen Sektionen eines OTU-Frames detailliert zu testen, ist in zwei entscheidenden Bereichen ebenfalls sinnvoll. Erstens besteht hiermit die Möglichkeit, das Design des FEC-Encoders und -Decoders zu verifizieren.

Zweitens kann der Anwender auf diese Weise Ausfälle oder Funktionsfehler des FEC-Algorithmus genauer einkreisen, sodass mangelhaft entworfene Encoder und Decoder für die 64 Unterzeilen schneller festgestellt werden können.

FEC-Generierung und I-Erkennung

Die Fehlergenerierung und -erkennung für den FEC-Bereich lässt sich für eine Reihe von Testlösungen in zwei Hauptbereiche unterteilen: Vordefinierte Tests der FEC-Funktionalität (korrigierbare, nicht korrigierbare und maximale Zahl korrigierbarer Fehler) sowie anwenderdefinierte (Experten-)Tests.

Einfügen korrigierbarer Fehler

In diesem Fall sollte der konfigurierte Prüfling in der Lage sein, eingefügte FEC-Fehler zu

korrigieren. Dieser einfache Test wird von Montageingenieuren benutzt, wenn es darum geht, in erster Näherung die korrekte Installation von NEs zu verifizieren. Bei einigen Testlösungen wird keine Fehlerangabe angezeigt, wenn der Prüfling den eingefügten FEC-Fehler korrigiert.

Die Implementierung erfolgt in diesem Fall dergestalt, dass Fehler in die erste Unterzeile der ersten OTU-Zeile eingefügt werden (Abb. 3). Da der RS-Code bis zu acht Symbolfehler je Unterzeile korrigieren kann, werden in der fraglichen Unterzeile genau acht Bytes mit Fehlern versehen. Invertiert werden, wie in Abb. 3 gezeigt, die Bytepositionen 240 bis 247 der Unterzeile.

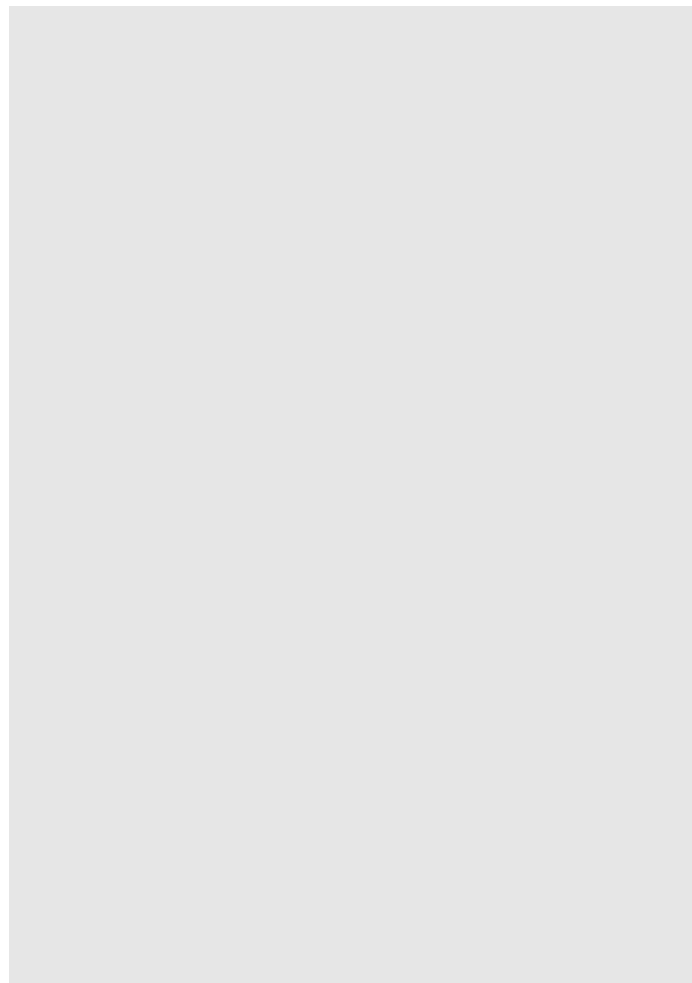
Die Auswahl der Positionen für die acht Byteinvertierungen wird mit der Vorgabe getroffen, dass bei Unfähigkeit des Decoders zum Korrigieren der Fehler die ursprünglichen Nutzdaten dennoch vollständig verfügbar gemacht werden können, da die Fehler ausschließlich in den FEC-Teil des OTU-Frames eingefügt werden.

Davon ausgehend, dass in der FEC-Implementierung 16 Encoder für alle OTU-Zeilen benutzt werden, würde es genügen, pro OTU-Zeile nur einen Encoder zu prüfen. Dies liegt daran, dass alle Encoder für eine Unterzeilen ein einheitliches Design besitzen und vor der Installation einem Maximum-Stress-Test unterzogen worden sein sollten.

Einfügen nicht korrigierbarer Fehler

Erforderlich ist es auch, die maximale Zahl der Fehler festzustellen, die der Prüfling korrigieren kann. Hier können die eingefügten Fehler vom Prüfling nicht mehr berichtigt werden. Beim Einfügen nicht korrigierbarer FEC-Fehler muss der Prüfling – sofern er korrekt funktioniert – einen akustischen Alarm auslösen.

Im Zuge einer solchen Prozedur werden in der ersten Unterzeile der ersten OTU-Zeile 16 Bytes invertiert. Dieser Ablauf ähnelt der Byteinvertierung für korrigierbare Fehler, jedoch werden im vorliegenden Fall alle 16 Bytes (240 bis 255) der Unterzeile invertiert.



Einfügen der maximalen Fehlerzahl

Diese Funktionalität erlaubt dem Benutzer das Einholen eines Pass/Fail-Kriteriums bezüglich der Fähigkeit des Prüflings zum Korrigieren von bis zu acht Symbolfehlern. Im vorliegenden Fall können einige Testlösungen die maximale Anzahl möglicher Byte- bzw. Symbolfehler einfügen, die der Prüfling zu korrigieren noch in der Lage sein muss. Anders ausgedrückt, wird das System bis an die Höchstgrenze belastet. Der Test fügt Fehler in den gesamten OTU-Frame ein.

Die Ausführung eines Maximum-Stress-Tests stellt sicher, dass die folgenden Testbedingungen erfüllt werden:

- ▶ Anwendung aller Kombinationen von Bytefehlermasken bei der Fehlereinfügung
- ▶ In jede Unterzeile werden maximal 8 Symbolfehler eingefügt (siehe Abb. 5)
- ▶ Alle 16 Unterzeilen aller vier OTU-Zeilen werden mit Fehlern versehen
- ▶ Alle möglichen Byte-Positionen des gesamten OTU-Frames enthalten Fehler

Um zu garantieren, dass alle Bytes des OTU-Frames mindestens einmal mit einem Fehler versehen werden, müssen die folgenden Parameter in jeder Test-Lösung intern bestimmt werden:

Startposition der Fehlereinfügung

Die Startposition der Fehlereinfügung wird von dem Moment ausgehend definiert, in dem die Insertion-Taste gedrückt wird.

Bestimmung der Bytefehlermaske

Die zufällige Erzeugung aller möglichen Bitkombinationen zwischen 0 x 01 und 0 x FF legt fest, welches Bit innerhalb eines Bytes invertiert wird. Denkbar ist beispielsweise die Maske F3 03 A1 B7 7B 04 AB 32, die die Invertierungspositionen der acht einzelnen fehlerbehafteten Bytes festlegt. Da sich diese Fehlermaske fortlaufend ändert, ergibt sich eine zufällige Sequenz generierter Symbolfehler.

Verteilung der Fehlerpositionen

Ist die Startposition der ersten acht Symbolfehler festgelegt, werden die Positionen der folgenden acht Bytefehler über den gesamten OTU-Frame verteilt. Dies garantiert eine pseudo-zufällige Verteilung der Fehler über den kompletten OTU-Frame.

Die beschriebene Testkonfiguration garantiert, dass nach einer etwa 3 ms langen FEC-Fehlereinfügung jedes Byte des OTU-Frames

TX	
Row select:	1010
Subrow select:	1100110011001100
Errored bytes per subrow: (1 ... 8 correctable, 9 ... 16 uncorr.)	7
Start position in subrow: (1 .. 255; 0=random)	154
Byte error mask: (00000001 .. 11111111; 0=random)	11011001

Abb. 5:

Beispiel für die spezifizierbaren Parameter in einer anspruchsvollen FEC-Testlösung

mindestens einmal mit einem Fehler behaftet war. Wenn der Prüfling auch unter maximaler Belastung korrekt funktioniert, werden sämtliche Fehler korrigiert. Das Signal, das an die Testlösung zurückgeführt wird, sollte dementsprechend keine Fehler aufweisen. Werden dagegen vom Prüfling Fehler ausgegeben, sind unter Umständen weitere Prüfungen mit anspruchsvolleren Testfunktionen vonnöten.

Fehlereinfügung und anspruchsvollere FEC-Funktionen

In einigen Lösungen können anspruchsvollere FEC-Funktionen tiefe Einblicke in die Implementierung eines FEC-Algorithmus ermöglichen.

Nützlich ist dies beispielsweise für die Forschung und Entwicklung, wenn es um den Entwurf und die Verifikation des FEC-Algorithmus geht, sowie zum schnellen Einkreisen von Ausfällen in Encoder- und Decoder-Designs. Im letzteren Fall werden außerdem weitere Funktionen genutzt, die die Grenzen der Korrekturfähigkeit eines Prüflings ausloten.

Anspruchsvollere FEC-Funktionen können auch Tests ober- und unterhalb dieser Grenzen durchführen. Im Einzelnen bieten sie dem Benutzer die Möglichkeit, genau anzugeben, welches Bit in welchem Byte einer OTU-Zeile und Unterzeile mit einem Fehler versehen werden soll.

Wie zuvor erwähnt, ist eine solche Funktionalität sinnvoll, um zu definieren, welches Bit an welcher Stelle eines OTU-Frames fehlerhaft sein soll. Abb. 5 macht deutlich, wie sich bestimmte Parameter spezifizieren lassen und wie die Benutzeroberfläche dieser Funktion in einer Testlösung gestaltet sein kann.

Im vorliegenden Beispiel bestünden die folgenden Auswahlmöglichkeiten:

- ▶ Festlegung von Zeile und Unterzeile
- ▶ Angabe der Startposition des Bursts in der Unterzeile

- ▶ Bestimmung der Anzahl fehlerbehafteter Bits je Unterzeile
- ▶ Angabe des zu invertierenden Bits in den ausgewählten Bytes.

Bei sorgfältiger Auswahl dieser Parameter ist es möglich, jedes einzelne Byte mit einem Fehler zu versehen und das resultierende Verhalten zu überprüfen.

B.05

Zusammenfassung

FEC-Lösungen sollten generell auf den Zweck ausgerichtet sein, dem Anwender die Möglichkeit zur eingehenden Prüfung von FEC-Funktionen zu geben. Gefordert ist außerdem Flexibilität, damit eine effiziente Nutzung in der ersten Entwicklungs- und Verifikationsphase sowie bei der Installation und Wartung von FEC-Systemen möglich ist.

Beitrag als PDF im Internet:

www.duv24.net
more @ click TK4B0503 

How to use

more @ click !

1. www.duv24.net
2. ,more@click'-Code eingeben
3. Anbieter kontaktieren – diskutieren – recherchieren