Implementação dos modulos Energy Dispersal e Playload CRC16 do protocolo DVB-RCS2: Prototipagem FPGA

1st Arthur Faria Campos
Programa de Engenharia Eletrônica
Universidade de Brasília - FGA
Brasília, Brasil
https://bitbucket.org/ArthurFariaCampos
arthur-fc@hotmail.com

2nd Gustavo Cavalcante Linhares
Programa de Engenharia Eletrônica
Universidade de Brasília - FGA
Brasília, Brasil
https://bitbucket.org/gustavoLinhares
gugacavalcante.10@gmail.com

***Resumo*—Este artigo descreve a implementação de hardware de dois módulos, *Energy Dispersal* e *CRC16*, em um Field-Programmable Gate Array (FPGA) reconfigurável da Xilinx usando VHDL Hardware Description Language. O *Energy Dispersal* foi projetado para embaralhar os bits afim de reduzir a probabilidade de seqüências de bits monotônicas. Enquanto o *CRC16* foi projetado para acoplar na saida do *Energy Dispersal* um valor de verificação curto, com base no restante de uma divisão polinomial de seus conteúdos.**

I. Introdução

Na geração atual de DVB (Digital Video Broadcast), é esperado que terminais do satélite sejam iterativos e capazes de transmitir dados com uma alta qualidade. Considerando o grande necessidade de tráfego e uma longa propagação de delay , uma técnica de acesso randômica é a solução para tal problema, esse acesso randômico é feito em DVB-RCS2 [1].

Todos os padrões DVB são desenvolvidos ou endossados pelo DVB Project, que é uma associação da indústria de mais de 200 fornecedores e usuários de equipamentos e software comprometidos com padrões técnicos abertos para serviços de dados e televisão digital. Embora os padrões tradicionais de transmissão de televisão digital permaneçam importantes, os padrões DVB de maior interesse e impacto futuro são aqueles que fornecem comunicações digitais de alta velocidade para IP (Internet Protocol) e acesso à Internet via satélite, como DVB-RCS2 e DVB-S2.

O DVB-RCS, e especialmente a sua segunda geração (DVB-RCS2), são de grande importância neste aspecto, suportando comunicações de satélite interactivas de alta velocidade (bi-direccionais) até 150 Mbps ou mais para transferências (Rx) e 75 Mbps ou mais para uploads (Tx) quando fornecido com capacidade de transponder adequada e configurações VSAT. Portanto, velocidades muito mais altas em conexões sem fio para assinantes são possíveis do que com redes móveis terrestres 4G atualmente.

II. Objetivos

O foco deste projeto e a implementação dos módulos Energy Dispersal e Playload CRC16 (Gen.CRC16) do receptor DVB-RCS2. A prototipagem foi realizada utilizando a FPGA da Xilinx *Xilinx Basys 3* e para a verificação do funcionamento dos módulos um modelo de referencia foi desenvolvido em Matlab.

Esses blocos fazem parte de um conjunto maior composto com os módulos da figura 1, esse do qual é responsável pela transmissão de dados através de pacotes [2].

Figura 1. Diagrama da transmissão de dados por pacotes

O módulo GEN.CRC16 faz conjunto com o módulo GEN.CRC32 o qual será desenvolvido posteriormente para compor o bloco GEN.CRC mencionado na Figura 1.

Outro meta buscada nesse projeto é integração desses módulos, pois além do funcionamento individual o conjunto total deve operar de forma correta, questões como timming e sincronismo farão parte dessa etapa.

III. Burst de transmissão

Cada burst de transmissão pertence a um tipo de transmissão atribuído a um tipo de conteúdo específico. Isso é determinado pelo especificação do tipo de transmissão no BCT [Tabela I].

Tabela I
Codificação do tipo de conteúdo de transmissão

Value	tx_content_type
0	Reserved
1	Logon Payload
2	Control Payload
3	Traffic and Control Payload
4	Traffic Payload
5 to 127	Reserved
128 to 255	User defined content

O formato e a sintaxe do payload são determinados pelo tipo de conteúdo e também o contexto de transmissão [Tabela II], enquanto o formato e a sintaxe são variantes de contexto.

Quatro tipos de conteúdo são definidos:

- **Logon** (formato de carga útil de quadro de variante de contexto não configurável)
- **Controle** (formato de payload de quadro de variante de contexto não configurável)
- **Tráfego / controle** (formato de carga configurável do quadro de transmissão da variante de contexto configurável)
- **Tráfego** (formato de payload configurável do quadro de transmissão da variante de contexto)

Tabela II
TAMANHO, FORMATO E SINTAXE DO PAYLOAD

Transmission Context	Logon Burst Payload	Control Burst Payload	Payload for Traffic
Transparent star, dedicated access	No Payload Label	No Payload Label	No Payload Label
Transparent star, Slotted Aloha access	6 byte; holding the RCST HID of the source	3 byte; concatenated source Group ID and Logon ID in sequence MSB to LSB	3 byte; concatenated source Group ID and source Logon ID in sequence MSB to LSB
Transparent star, CRDSA access	8 byte; Concatenated 6 byte holding the RCST HID of the source and 2 byte CRDSA tag, in sequence MSB to LSB	5 byte; Concatenated source Group ID, Logon ID and CRDSA tag, in sequence MSB to LSB	5 byte; Concatenated source Group ID, source Logon ID and CRDSA tag, in sequence MSB to LSB
Transparent mesh overlay	not applicable	not applicable	2 byte for transmitter identification
Regenerative mesh	6 byte; holding the RCST HID of the source	3 byte; concatenated source Group ID and Logon ID in sequence MSB to LSB	2 byte for receiver identification

A. *Definições*

Assim, para modelarmos um modulo inicial utilizaremos 4 tamanhos de Frame PDU. Assim como uma entrada para o tipo de frame [Tabela III].

Tabela III
TAMANHOS DE FRAME

Tipo	Bytes
PRBS_SHORT_FRAME	8
PRBS_NORMAL_FRAME	12
PRBS_MEDIUM_FRAME	24
PRBS_LARGE_FRAME	32
PRBS_MAX_FRAME_DVBRCS2	32

Outra importante consideraçao é o Frame PDU que está sendo enviado em rajadas de 32bits, **Frame maximo**, para o energy dispersal juntamente com o tipo de frame.

Os campos de burst serão emitidos em ordem de bits começando com o bit considerado mais significativo(MSB) e terminando com o menos significativo (LSB).

IV. ENERGY DISPERSAL

Afim de cumprir os Regulamentos de Radiocomunicações ITU e para garantir transições binárias adequadas, o fluxo de bits em um Burst deve ser misturado para reduzir o probabilidade de sequências de bits monotônicas.

Uma sequência de pseudo aleatória de números binários (PRBS - Pseudo-Random Binary Sequence) deve ser usada conforme especificado pela expressão polinomial [1], tal expressão surge de descrição matemática do diagrama de blocos da figura 2 e também da quantidade de bits utilizada [3].

$$1 + x^{14} + x^{15} \tag{1}$$

Os dados são embaralhados usando o registrador Linear Feedback Shift Register (LFSR) de 15 registros mostrado na figura 2 para distribuir a distribuição de uns e zeros.

O gerador realiza a adição do módulo-2 dos dados com a sequência pseudo aleatória. A cada novo Burst o LFSR é resetado para o conteudo inicial da tabela IV.

Tabela IV
VALOR INICIAL DO RANDOMIZER

Shift register	SR1	SR2	SR3	SR4	SR5	SR6	SR7	SR8	SR9	SR10	SR11	SR12	SR13	SR14	SR15
Bit value	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0

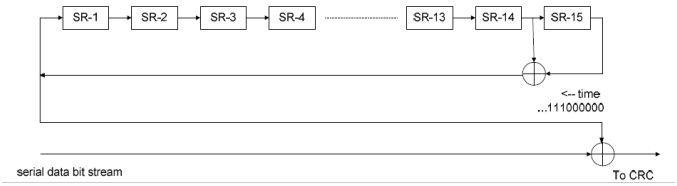

Figura 2. Linear Feedback Shift Register

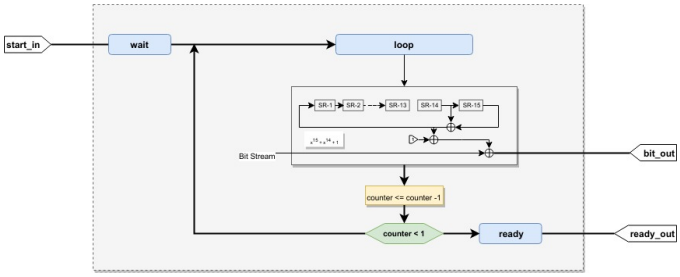

Figura 3. Energy Dispersal Projetado

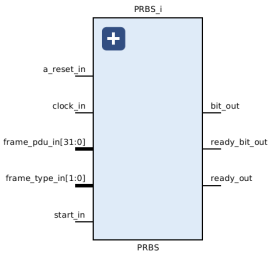

Figura 4. RTL

A. *Simulações*

Para realizar a simulaçoes foram gerados valores aleatorios em Matlb e convertidos para binario, assim como também os

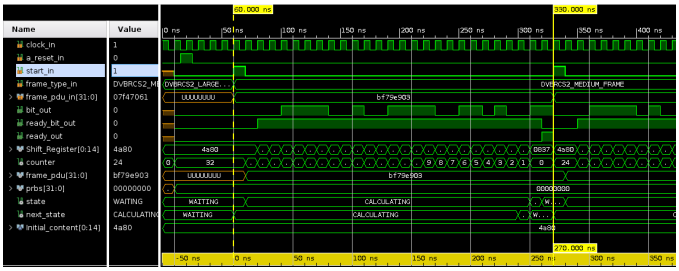

Figura 5. Simulação (1/2)

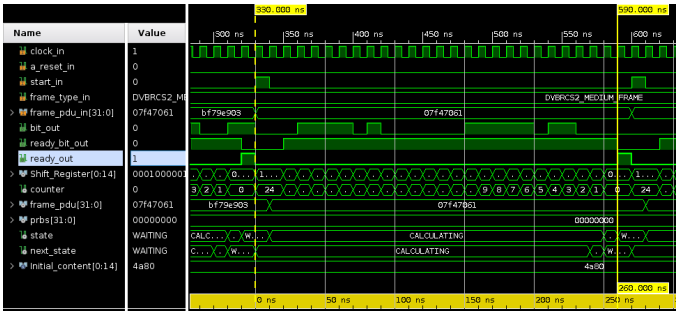

Figura 6. Simulação (2/2)

tipos de frame. Realizamos então o input deses valores na simulaçao do vivado para calcular a latencia e o Throughput.

A latência irá ser alterada para cada tipo de frame, mas para um frame Médio(24 bits) é de:

$$latencia = 260ns$$

O bloco entao apresenta um latetncia equivalente a:

$$clock \times (framebytes + 2)$$

.

B. Recursos

Para realizar a simulaçoes foram gerados valores aleatórios em Matlab e convertidos para binário, assim como também os tipos de frame. Realizamos então o input deses valores na simulaçao do vivado para calcular a latencia e o Throughput.

Resource	Utilization	Available	Utilization %
LUT	27	20800	0.13
FF	64	41600	0.15
IO	40	106	37.74

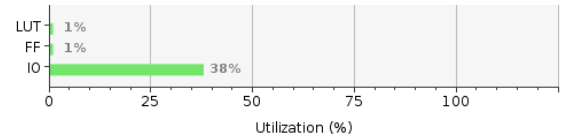

Figura 7. Recursos do Energy Dispersal

V. CRC - Cyclic Redundancy Check

Existem vários problemas que podem causar mudanças de bits na transmissão de dados. Com o intuito de reduzir ao máximo o número dessas mudanças acidentais em dados binários surgem os códigos de detecção de erros. Um desses códigos de detecção de erros faz parte dos blocos implementados neste projeto, o Cyclic Redundancy Check (CRC).

O objetivo do CRC é calcular uma divisão polinomial e concatenar o resto dessa divisão ao dado transmitido. O divisor do polinômio (key) e o numero de bits dos dados são pré determinado pelo protocolo de comunicação. Assim quando o dado for recebido essa mesma divisão será calculada, e caso o resto for igual a zero pode-se confirmar que o dado não sofreu nenhum tipo de dano durante a transmissão.

Como os dados são transmitidos via serial, o calculo do polinômio é feito através de uma LFSR. Dado o nome do módulo CRC16 a nossa key possui 16 bits e é dada pela seguinte expressão [2].

$$x^{16} + x^{15} + x^2 + 1 \quad (2)$$

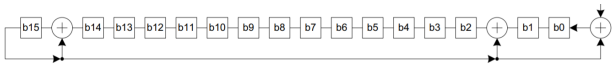

Figura 8. LFSR para o cálculo- $x^{16} + x^{15} + x^2 + 1$

O diagrama da LFSR é mostrado na figura 9, onde os dados entram na soma antes do registrador D0 e após todo o número passar pela LFSR, o resto da divisão será dado pelo valor dos registradores D0 a D15.

A. CRC16 - Resultados de Implementação

Após a implementação do diagrama da fig 9 a seguinte tabela de de utilização foi obtida, levando em consideração que não foi feita a implementação fora de contexto, ou seja, esses pinos de IO devem ser ignorados.

Utilization Post-Synthesis | **Post-Implementation**

Graph | **Table**

Resource	Utilization	Available	Utilization %
LUT	21	53200	0.04
FF	39	106400	0.04
IO	27	200	13.50
BUFG	1	32	3.13

Figura 9. Tabela de utilização do CRC16

Com relação a latência e throughput, esses fatores variam de acordo com o tamanho do dado de entrada. Melhores análises serão feitas futuramente já que o funcionamento do módulo já esta correto.

VI. Considerações Finais

Muitos conceitos novos foram introduzidos, uma melhor visão do projeto da autotrac foi alcançado e o entendimento sobre o funcionamento dos módulos foi dado. Os vhds não serão o grande desafio desse projeto assuntos como documentação e integração de módulos tomarão a maior parte no tempo de desenvolvimento, por tais motivos, uma melhor comunicação com os envolvidos juntamente com cronograma serão fundamentais para uma conclusão efetiva do projeto.

Referências

[1] A. Meloni and M. Murroni, "Random access in DVB-RCS2: design and dynamic control for congestion avoidance," *CoRR*, vol. abs/1501.06361, 2015. [Online]. Available: http://arxiv.org/abs/1501.06361

[2] E. B. Union., "Digital vidideo broadcasting (dvb); secocond generation dvb interactive satellite system (dvb-rcs2); part 2: Lower layers for satellite standard." p. 130, 2014.

[3] H. Okawara's, "Dsp-based testing – fundamentals 50 prbs (pseudo random binary sequence)," *ADVANTEST Corporation*, p. 02, 2013. [Online]. Available: https://www.advantest.com/documents/11348/3e95df23-22f5-441e-8598-f1d99c2382cb