

DESENVOLVIMENTO DE MÓDULO DE COMUNICAÇÃO SPACEWIRE USANDO PLATAFORMA LEON - SIMULADOR DE DADOS PARA O SATÉLITE PLATO

Rafael Corsi Ferrão¹ ; Sergio Ribeiro Augusto²

¹ Aluno de Iniciação Científica da Escola de Engenharia Mauá (EEM/CEUN-IMT);

² Professor da Escola de Engenharia Mauá (EEM/CEUN-IMT).

RESUMO.

Este projeto de iniciação científica consiste no desenvolvimento de uma plataforma EGSE (Electrical Ground Segment Equipment) que fará a simulação da N-FEE (Normal-Front End Electronic) do satélite PLATO (PLAnetary Transits and Oscillations of Stars) . Tal EGSE será utilizado para a execução de testes na N-DPU(Normal-Digital Processor Unity) do satélite e também para a validação da comunicação entre ambos dispositivos. A plataforma proposta deverá enviar imagens artificiais carregadas de um computador e transmitir tais imagens obedecendo as especificações propostas pela equipe da LESIA (Laboratoire d'études spatiales et d'instrumentation en astrophysique) do Observatório de Paris. Os protocolos utilizados para a comunicação entre a N-FEE e a N-DPU são: o protocolo espacial de alta velocidade SpaceWire e o protocolo RMAP . O sistema foi parcialmente implementado em VHDL e foi proposto uma nova arquitetura utilizando a biblioteca VHDL GRLIB que possui em seu núcleo o processador LEON3 .

Introdução

O satélite PLATO (Plato, 2011) é um projeto do programa *Cosmic Vision* da ESA (*European Space Agency*) que se encontra na fase B (definição). Trata-se de um grande fotômetro composto por dezenas de espelhos, cuja finalidade é a detecção de milhares de novos planetas através do método de detecção por trânsito planetário (eclipse). Para cada sistema planetário detectado, um estudo sismológico de alta precisão será executado na estrela central, propiciando a caracterização completa e muito precisa dos parâmetros físicos do sistema.

O instrumento é composto de 32 câmeras dedicadas designadas por normais (N-FEE), termo que identifica sua funcionalidade na arquitetura e 2 câmeras rápidas utilizadas na malha de controle de altitude (F-FEE). Cada câmera normal possui 4 CCDs (Charge Coupled Device) resultando em um conjunto de 128 CCDs, conforme representação da Figura 1.

As câmeras terão suas posições ajustadas visando obter uma maior abertura. Cada N-FEE é conectado a uma N-DPU (Thales, 2011) através de dois canais SpaceWire (ESA, 2008), possibilitando assim uma redundância na comunicação e uma menor utilização do link.

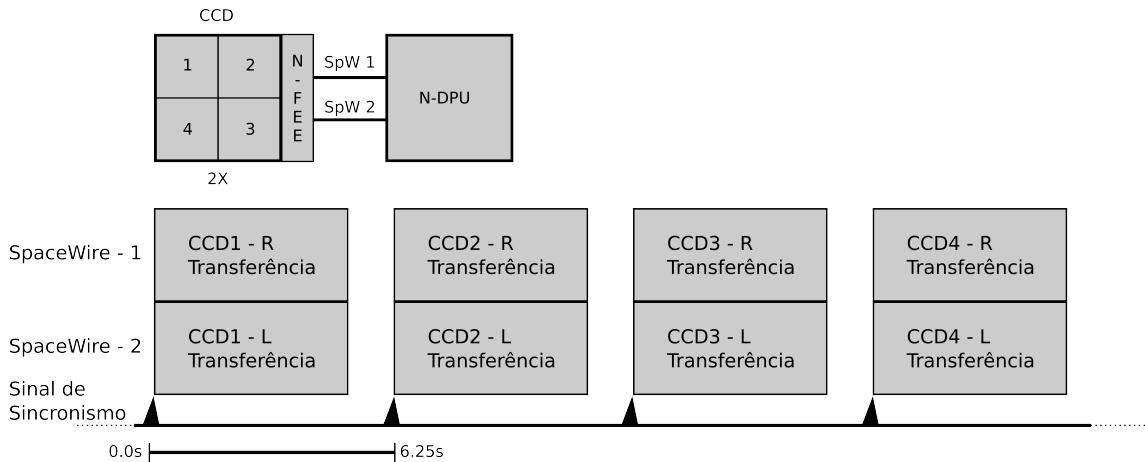


Figura 1: Carta de tempo, leitura e envio das imagens - LESIA

Material e Métodos

Hardware

Foi adquirido para a execução da arquitetura uma placa GR-PCI-XC5V (Aeroflex Gaisler, 2008) através de recursos do INESpaço (Instituto Nacional de Tecnologias e Ciências do Espaço) CNPQ-INCT-AEB sob coordenação do programa Astrobiologia. A placa foi concebida especialmente para apoiar o desenvolvimento e prototipagem rápida de sistemas baseados no processador LEON. Também foram adquiridos o simulador GRSIM (Gaisler, 2010), capaz de fazer simulações da arquitetura do processador LEON3 com suporte a múltiplos núcleos de processamento, e o debugador GRMON (Gaisler, 2010), o qual possibilita o acesso a informações do processador. Adquiriu-se também dois cabos SpaceWire de 10m cada para utilização em laboratório, assim como um SpaceWire-USB Brick da Star-Dundee desenvolvido para verificação e teste de canais SpaceWire. Posteriormente adquiriu-se uma placa de desenvolvimento DE4 da Altera (Altera, 2011) para a futura implementação de uma arquitetura dedicada ao EGSE.

Comprou-se também dois IP Cores SpaceWire (*codec*): um do fabricante SpaceWire UK (UK, 2011) que fornece o código VHDL do produto e não utiliza tecnologias específicas de fabricantes de FPGAs, possibilitando assim a análise e modificação de seu código e a implementação em diferentes tipos de pastilhas, e outro IP *core* fornecido pela LESIA. Este último é dedicado a pastilhas Xilinx (Xilinx, 2011) e disponível somente na forma de *netlist* (código já copilado), impossibilitando assim alterações analisáveis de código.

Resultado e Discussões

N-FEE

Os seguintes princípios de funcionamento da N-FEE foram propostos pela equipe CNES (*Centre National d'Études Spatiales*) e foram tomados como referência neste pro-

jeto:

- Cada N-FEE é conectado a uma N-DPU através de dois canais SpaceWire, sendo que é possível a utilização simultânea ou individual dos canais. Tal modo de operação é escolhido pela N-DPU, assim como qual canal utilizar caso esteja utilizando um único canal. A velocidade dos canais são de 100 Mbps implicando em um carregamento de 80 % no caso de um único canal e de 70 % quando utilizado dois canais (Thales, 2011).
- A transferência da imagem sobre o canal SpaceWire é feita utilizando-se o protocolo RMAP (ESA, 2008) com o comando de escrita sem reconhecimento(*no-ack*) e sem verificação(*no-verify*). Envia-se meia linha por pacote (2255 pixels + 25 prescan pixels).
- Um sinal externo de sincronismo é fornecido a placa via um simples sinal (com cadência de 6,25s) e indica o início da descarga da imagem no link SpaceWire. A ordem desta descarga é descrita pela Figura 1.
- Quando recebido o sinal de sincronismo deve-se transmitir antes da imagem um TimeCode (TimeCode é o meio de propagação de tempo utilizado no protocolo SpaceWire) indicando qual CCD está sendo transmitido (0,1,2 ou 3). Essa informação será utilizada pela N-DPU para sincronização interna já que a mesma não possui acesso direto ao sinal de sincronismo.
- O TimeCode deve ser enviado pelos dois canais, sendo a N-DPU responsável por escolher em qual canal irá amostrar.
- São adicionados 16 bits a cada dado (meia linha de um CCD) contendo um contador de 6,25s, onde os 2 mais significativos bits indicam a qual CCD pertence a linha sendo transmitida (0,1,2 ou 3). Tal recurso possibilita assim detectar perdas de pacotes na transmissão.

O protocolo SpaceWire

Spacewire é um padrão de comunicação de alta velocidade (200 Mbps típico) voltado para aplicações espaciais. O mesmo implementa um link diferencial LVDS (*Low voltage differential signalling*) (National, 2008) serial bi-direcional ponto a ponto de alta capacidade, permitindo uma eficiente troca de informações com dispositivos de sensorialmente embarcados, vindo de encontro com os requisitos necessários para aplicações espaciais e de aviação. Em nível de implementação o protocolo SpaceWire pode ser implementado em diferentes tipos de tecnologias, incluindo FPGAs (*Field Programmable Gate Array*). O padrão define basicamente o encapsulamento de pacotes de dados (camada física e de enlace do modelo OSI), não tratando as camadas superiores de comunicação. Baseado em parte no padrão IEEE-1553 (ALta, 2007) e LVDS, foi proposto pela ESA, passando, desde então, por contínuas implementações. Os objetivos do protocolo são:

- facilitar a construção de sistemas de processamento de dados *on-board* de alto desempenho;
- auxiliar na redução dos custos de integração de sistemas;
- coleta e armazenamento de dados dos instrumentos; assim como telemetria.

SpaceWire faz uso da codificação Data-Strobe (DS). Este esquema de comunicação codifica a mensagem (dado) com o *clock* (sinal de relógio), para que este *clock* possa ser recuperado no receptor através de uma simples porta XOR entre o sinal dado e o DS. A vantagem desta codificação é obter uma tolerância de quase 1 tempo de bit, comparado com 0,5 para a codificação simples (Esa, 2008).

No protocolo SpaceWire são definidos os seguintes tipos de caracteres:

- Caracteres do tipo dado: são formados por nove bits, sendo o bit menos significativo transmitido primeiro. Um bit de paridade é adicionado a transmissão cobrindo os 8 bits anteriores. Um bit é utilizado como flag para definir se a mensagem é do tipo comando ou do tipo dado.
- Caracteres de controle: são formados por um bit de paridade, um de controle de dado (colocado em nível lógico '1' para indicar que é um caractere de controle) e dois *data-caracteres*.

Um caractere de Flow Control (FCT) é enviado sempre que houver espaço na memória de recebimento. O receptor incrementa um contador a cada FCT recebido e decrementa quando enviado um caractere de dado. É permitido um acumulo máximo de 56 FCT, caso contrário o link deve entrar em estado de erro.

Um caractere do tipo NULL é enviado sempre que o link se encontra em estado ocioso, mantendo assim o link ativo e possibilitando a detecção de rompimentos do canal de comunicação.

O protocolo RMAP

O protocolo RMAP (*Remote Memory Access Protocol*) foi desenvolvido pela comunidade europeia (científica e industrial) como complemento para o protocolo SpaceWire na maneira a possibilitar a leitura e gravação de dados em memória pertencentes a nós remotos de um canal SpaceWire. O seu principal propósito é a de configuração de redes SpaceWire e a coleta de informações e dados de diferentes nós.

As operações definidas no protocolo RMAP são : leitura, gravação, leitura-modificação-gravação. A operação de gravação fornece um meio de um nó (iniciador) gravar em outro nó zero ou mais dados em uma área específica da memória. A operação de leitura é um meio do iniciador ler zero ou mais dados de uma área específica da memória remoto. A operação de ler-modificar-gravar possibilita o iniciador ler uma zona da memória remota e fazer modificações na mesma área.

O comando de gravação pode ser executado com reconhecimento (*ack*), onde uma resposta indicando se o comando foi executado com sucesso ou não (*erro/status*) é enviado do

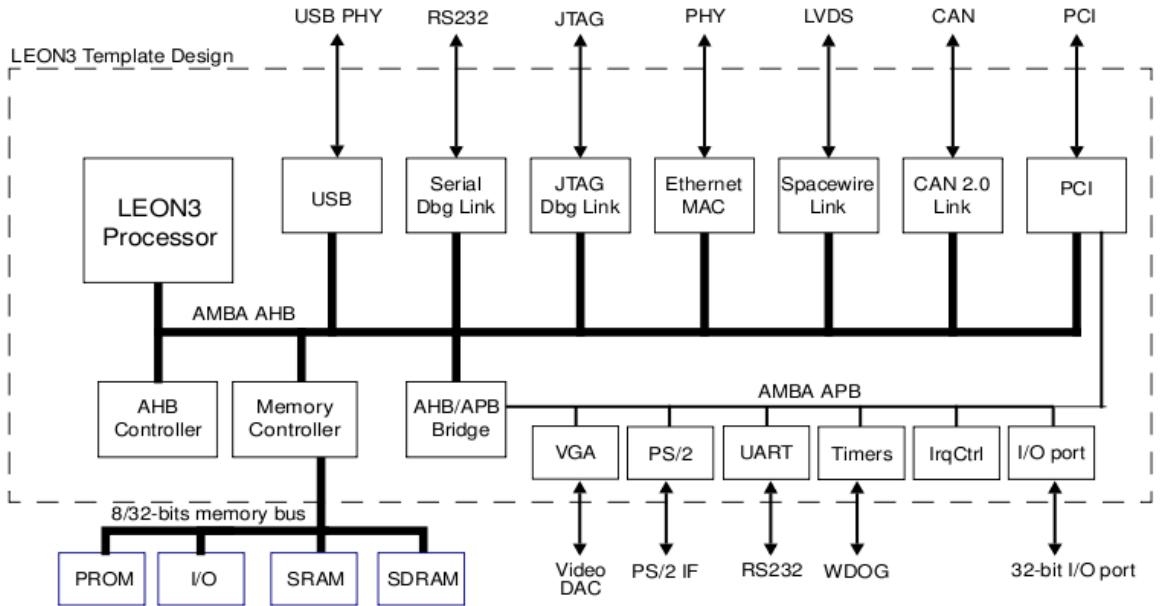


Figura 2: Esquema de utilização do GRLIB

nó remoto. Pode-se também escolher pela verificação do dado antes do mesmo ser gravado na memória do alvo, ou seja, o nó alvo recebe a informação e executa um CRC (*Cyclic Redundancy Check*) (Michael e Frank, 2005) na parte do pacote pertencente ao dado. Caso os valores do CRC sejam iguais o dado é então gravado na memória caso o contrário, a informação do pacote é descartada. Pode-se fazer combinações de comandos, como por exemplo um comando de gravação com verificação e reconhecimento (*ack+CRC*).

Foi escolhido para a transferência de imagem entre o N-FEE e a N-DPU o comando de escrita sem reconhecimento e sem verificação. Sem reconhecimento pois o N-FEE não possui as informações enviadas em memória e uma mensagem perdida não poderia ser recuperada, e sem verificação para diminuir o peso e custo da aplicação já que seria necessário um banco de memórias para fazer o buffer da mensagem e execução do cálculo do CRC.

Processador LEON3

O LEON3 é um processador de 32-bits baseado na arquitetura SPARC V8 (Gaisler, 2008), desenvolvido para aplicações embarcadas, combinando alta performance com baixa complexidade e baixo consumo energético. Distribuído através do VHDL GRLIB IP (propriedade intelectual) sobe licença GPL (General Public License) contém além do processador, módulos como: controlador de memória, interface PCI, USART, Ethernet MAC. Um exemplo de aplicação das bibliotecas presentes no GRLIB pode ser visto na Figura 2.

O GRLIB IP foi desenvolvido para ser uma plataforma modular e flexível, onde todos os módulos compartilham o mesmo barramento AMBA2.0 (Arm, 1999) facilitando assim a criação de novos projetos.

O processador LEON3 implementa uma arquitetura Harvard (Gailser, 2010) com ins-

truções e *cache* de dados separados. Possui divisão e multiplicação em hardware e uma unidade de processamento de ponto flutuante pode ser adicionada. Contém também uma unidade de gerenciamento de memória que possibilita o mapeamento de 32-bits de endereço virtual e 32-bits de memória. Pode funcionar com múltiplos processadores, os quais podem compartilhar uma mesma memória.

Desenvolvimento

Uma primeira plataforma foi desenvolvida inteiramente em VHDL, sendo é capaz de encapsular uma série de informações pré definidas em um pacote RMAP (sem reconhecimento e sem verificação) e enviar tais informações via um canal SpaceWire utilizando um dos *codecs* adquiridos. Tal plataforma (figura 3) não possibilita o carregamento da imagem no dispositivo e foi criada por dois motivos: para a melhor integração entre o LEON e o *codec* SpaceWire e devido ao atraso nas entregas dos materiais que impossibilitou o início da implementação com o processador LEON.

A segunda plataforma proposta (figura 4) utiliza o processador LEON3 com comunicação UART (*Universal Asynchronous Receiver/Transmitter*). No início foi cogitado a utilização do protocolo USB2.0 porém tal módulo não é fornecido na parte gratuita do GRLIB IP. O empacotamento da mensagem no protocolo RMAP é feito via o módulo RMAP de gravação, para tanto, as entradas e saídas do módulo foram mapeadas em memória do processador, possibilitando assim o acesso do LEON ao módulo.

Módulo RMAP de gravação

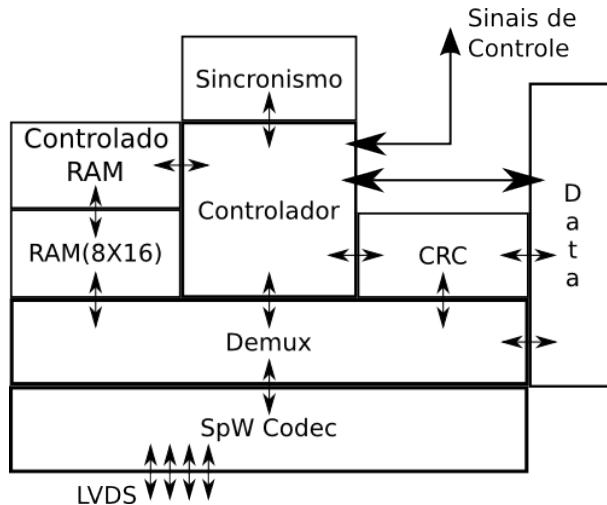


Figura 3: Módulo RMAP de gravação

O módulo criado em VHDL possui a arquitetura da figura 3 e foi criado para funcionar em modo *stand-alone*. Alguns parâmetros do protocolo RMAP já estão pré definidos e o cabeçalho da mensagem possui o formato da tabela 1, podendo ser alterados externamente por comandos ao módulo. Possui os seguintes modos de operação :

Campo	Valor(hexa)	Observação
Logicall Address	FE	Valor padrão
Identifier Field	01	Identificação do protocolo RMAP 01(packet type)
Instruction Field	60	1000(write non ack non verify) 00(no reply field)
KEY Field	00	A ser definido
Initiator Logical Address	FE	Valor padrão
Trasaction Identifier Field (MSB)	00	É incrementado a cada nova mensagem
Trasaction Identifier Field (LSB)	00	-
Extend Address Field (MSB)	00	Endereço na memória; A ser definido
Address Field	00	A ser definido
Address Field	00	A ser definido
Address Field	00	A ser definido
Address Field	00	A ser definido
Address Field (LSB)	00	A ser definido
		16*2255 (meia linha) +
Data Lenght field (MSB)	8D	25 (pré-scan) + 16 bits (identificação) = h8D19
Data Lenght field (LSB)	19	

Tabela 1: Cabeçalho RMAP para gravação na N-DPU

- Modo normal : Quando recebido o sinal de sincronismo, o módulo começa a funcionar sem intervenção do processador. Envia-se o cabeçalho com as informações pré definidas e inicia-se a transferência dos dados. Todo o gerenciamento é feito no próprio bloco.
- Modo Incremental de atualização: Neste modo as informações do cabeçalho são atualizadas uma a uma em modo incremental, sem a necessidade de endereçamento externo;
- Modo Aleatório de atualização : Aqui uma informação do cabeçalho é modificada e o endereçamento externo é necessário.

Pode-se observar no anexo 6 a simulação do funcionamento do módulo. Os sinais de controle do módulo (*clock, reset, sync, operation mode, enable data*) foram sintetizados manualmente, de forma a simular uma interface com um controlador externo.

Arquitetura - LEON

A segunda arquitetura (figura 4) utiliza o processador LEON3 na placa GR-PCI-XC5V, com comunicação UART e com o módulo RMAP de escrita mapeado em memória. É utilizada para tanto a biblioteca APBUART (Gaisler, 2010) a qual é um módulo UART conectado ao AMBA AHB (*Advanced High Bus*). A imagem é enviada à RAM (Memória de acesso aleatório) da placa através do módulo UART. Essa forma de comunicação

impossibilita a leitura dinâmica de imagens (nova imagem a cada sincronismo) devido a velocidade de comunicação entre o computador e a plataforma, pode-se futuramente implementar uma comunicação PCI (*Peripheral Component Interconnect*) que possui a velocidade necessária (mínimo de 100Mbps) para o funcionamento em modo dinâmico.

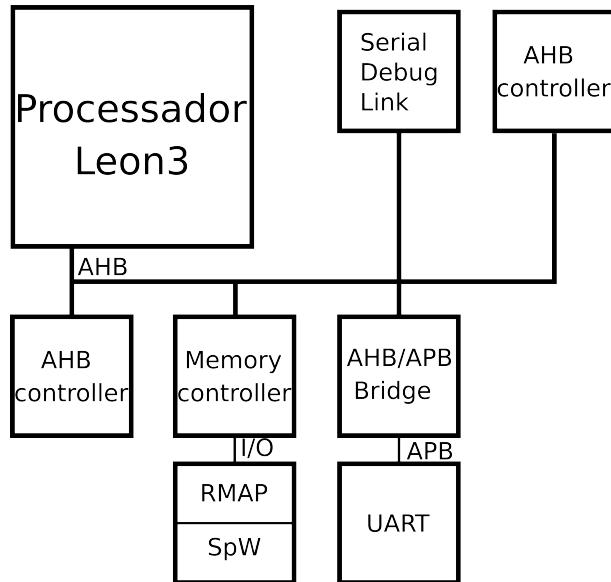


Figura 4: Arquitetura proposta utilizando o processador LEON3

O processador é responsável pela gerência das comunicações PC, Placa, N-DPU (figura 5). O sinal de sincronismo externo é mapeado através uma interrupção no processador que desencadeia todo o processo de envio da imagem.

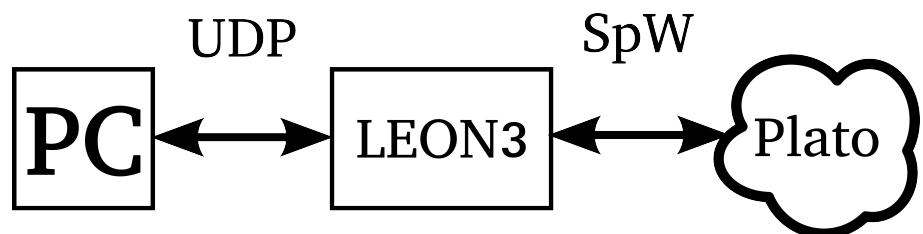


Figura 5: Modelo geral de comunicação entre o Computador e o Satélite

Conclusões

Este trabalho de iniciação científica estudou novas tecnologias aeroespaciais, suas aplicações e implementações. Aqui, o desafio é a escolha e a integração das ferramentas gratuitas disponíveis. Pretende-se estabelecer bases para a homologação de projetos que envolvam o uso de tecnologias aeroespaciais (LEON, RMAP e SpaceWire).

O atraso da entrega dos materiais dificultou o desenvolvimento da arquitetura inicialmente prevista. Optou-se pelo o desenvolvimento do módulo RMAP em VHDL já que as ferramentas de testes e compilação estavam disponíveis para uso.

Obteve-se o resultado esperado com o módulo desenvolvido, porém é necessário o desenvolvimento de uma nova parte que execute a operação de leitura do protocolo RMAP. Também é preciso após um estudo mais aprofundado do GRLIB IP, a implementação de uma comunicação mais rápida entre o EGSE e o computador, a fim de obter um melhor comportamento dinâmico do N-FEE.

Referências Bibliográficas

- Aeroflex Gaisler (2008) GR-PCI-XV5V Development Board. Disponível em: <http://www.gaisler.com/cms>. Acessado em Agosto de 2010.
- Alta Data Technologies LLC (2007) MIL STD 1553 Tutorial and Reference. Disponível em: <http://www.altadt.com/support/tutorials/mil-std-1553-tutorial/>. Acessado em Abril de 2010.
- Altera (2011) FPGA CPLD and ASIC from Altera. Disponível em: <http://www.altera.com>. Acessado em Outubro de 2011.
- ARM Limited (1999) AMBA™ Specification. Rev 2.0. Disponível em: <http://www.arm.com/products/system-ip/amba/amba-open-specifications.php>. Acessado em Abril de 2010.
- ESA Requirements and Standards Division (2008) SpaceWire Links, nodes, routers and networks – ECSS-E-ST-50-12C. Disponível em: <http://www.esa.int/esaCP/index.html>. Acessado em Julho de 2010.
- ESA Requirements and Standards Division (2008) SpaceWire protocols – ECSS-E-ST-50-11. Disponível em: <http://www.esa.int/esaCP/index.html>. Acessado em Julho de 2010.
- Gaisler (2010) GRLIB IP Core User's Manual. Disponível em: <http://www.gaisler.com/cms>. Acessado em Outubro de 2010.
- Gaisler (2010) GRSIM Users manual 1.0. Disponível em: <http://www.gaisler.com/cms>

Acessado em Outubro de 2010.

Gaisler (2011) GRMON Users Manual 0.85. Disponível em: <http://www.gaisler.com/cms>
Acessado em Janeiro de 2011.

Gaisler (2011) GRMON Users Manual 0.85. Disponível em: <http://www.gaisler.com/cms>
Acessado em Fevereiro de 2011.

Michael E. Frank LB. (2005) A Systematic Approach to Building High Performance,
Software-based, CRC Generators. Disponível em:
http://www.di.uoa.gr/en/research_publ2.php. Acessado em Fevereiro de 2011

National Semiconductor (2008) LVDS Owner s Manual - Including High-Speed CML and
Signal Conditioning. Disponível em: <http://www.national.com/>. Acessado em
Janeiro de 2011.

Plato (2011) Disponível em: <http://www.lesia.obspm.fr/perso/clause-catala/plato-web.html>,
Acessado em Janeiro de 2011.

Thales (2011) DPU – FEE interface requirement. Paris. Documento interno LESIA.

UK SpaceWire (2011). Disponível em: <http://www.spacewire.co.uk/> Acessado em Janeiro de
2011.

Xilinx (2011) FPGA, CPLD, and EPP Solutions from Xilinx, Inc. Disponível em:
www.xilinx.com. Acessado em Outubro de 2011.

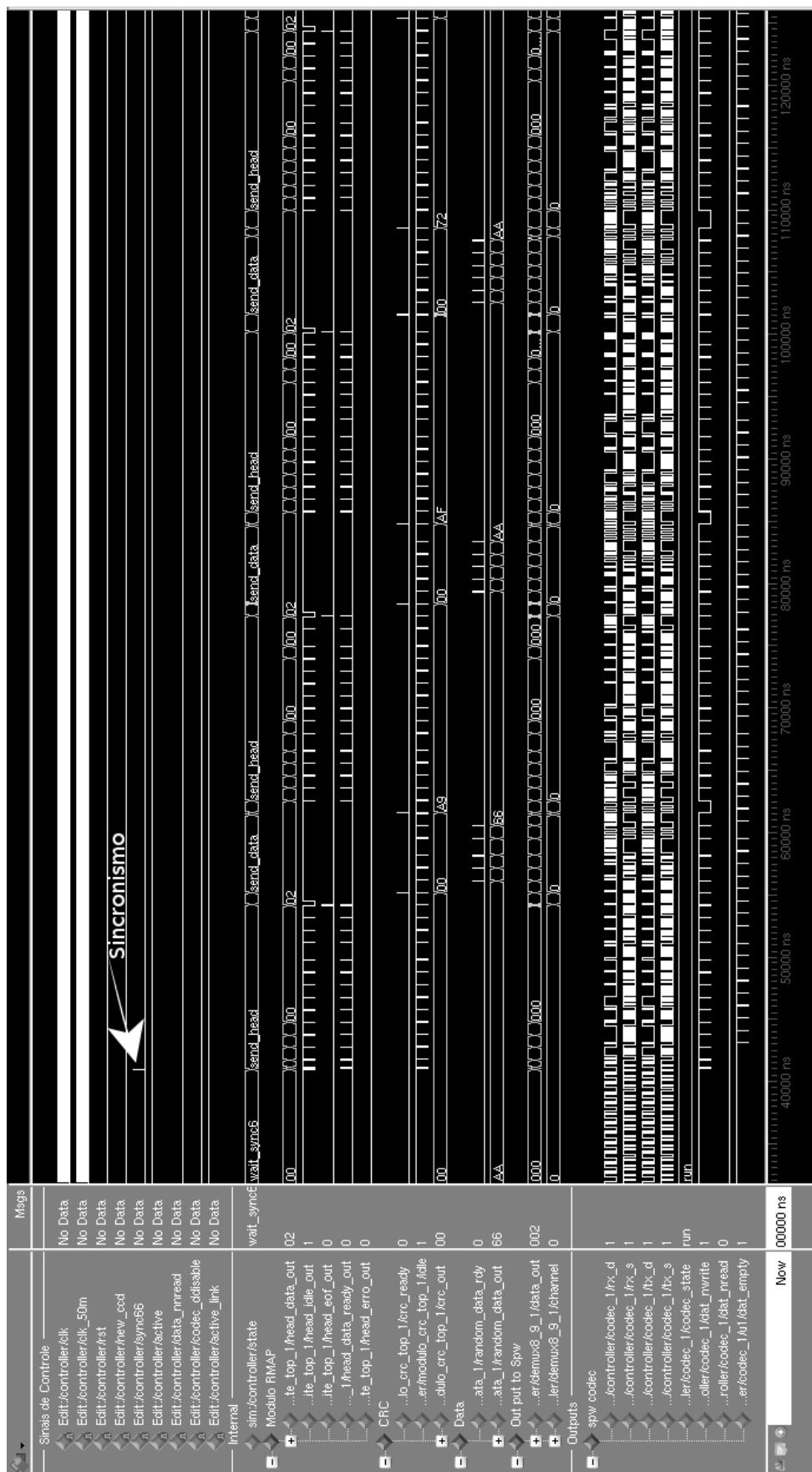


Figura 6: Simulação do módulo RMAP executada no simulador MODELSIM