



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

TRANSMISSOR REALIZADO EM SOFTWARE

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA (PIBIC/CNPq/INPE)

Bruno Augusto Ferreira Vitorino (UFRN, Bolsista PIBIC/CNPq)
E-mail: bafvufrn@gmail.com

Dr. Fernando Rangel de Sousa (DEE/UFRN, Orientador)
E-mail: frangel@dee.ufrn.br

Manoel Jozeane Mafra de Carvalho (CRN/INPE, Orientador)
Email: manoel@crn.inpe.br

Julho de 2008

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO (INTRODUÇÃO, OBJETIVOS DO TRABALHO)

CAPÍTULO 2 – SISTEMA BRASILEIRO DE COLETA DE DADOS

- 2.1 Estrutura do Sistema de Coleta de Dados.
- 2.2 Plataformas de Coleta de Dados (PCD's).
- 2.3 Transmissores e técnicas de modulação usadas.

CAPÍTULO 3 – CORDIC (ALGORITMO PARA PROCESSAMENTO DE SINAIS)

- 3.1 Estrutura do Algoritmo CORDIC.
- 3.2 Modelo de Referência em Linguagem C.
- 3.3 Bloco Funcional CORDIC (Circuito RTL)

CAPÍTULO 4 – MODULADOR DIGITAL BASEADO NO CORDIC

- 4.1 Esquema para modulação baseado no módulo CORDIC
- 4.2 Modelo de Referência para Modulador (Linguagem C)
- 4.3 IP Core Modulador (Circuito RTL)
- 4.4 Modelo Experimental (Prototipação em FPGA)

CAPÍTULO 5 – RESULTADOS E ANÁLISES DO MODULADOR

- 5.1 Resultados de simulação.
- 5.2 Resultados experimentais do Modulador.

CAPÍTULO 6 – COMUNICAÇÃO SERIAL (RS-232) EM VHDL

- 6.1 Interface Serial Assíncrona (UART)
- 6.2 Projeto de UART em VHDL.
- 6.3 Modelo de testes da interface serial.

CAPÍTULO 7 – CONCLUSÕES E TRABALHOS FUTUROS

REFERÊNCIAS

CAPÍTULO 1 – INTRODUÇÃO

Este projeto de iniciação científica tem como propósito implementar um transmissor para comunicação com satélites do Sistema Brasileiro de Coleta de Dados do INPE (Instituto Nacional de Pesquisas Espaciais). O transmissor a ser desenvolvido será acoplado a Plataformas de Coleta de Dados Meteorológicos do INPE em substituição aos transmissores mais antigos que são grandes e bem mais robustos do que o proposto neste projeto.

O transmissor está sendo desenvolvido utilizando técnicas de comunicação digitais e o conceito de circuitos reconfiguráveis, que se baseiam em FPGA (*Field Programmable Gate Arrays*), circuitos lógicos que permitem o desenvolvimento das mais diversas tarefas e são programáveis através de HDL's (*Hardware Description Language*). Existem várias linguagens para descrição de hardware, as mais usadas são Verilog e VHDL, neste projeto foi usada a linguagem VHDL.

O transmissor possui um módulo digital, o modulador, e circuitos analógicos de Rádio Frequência subsequentes ao modulador. O módulo digital já foi desenvolvido e prototipado em FPGA e encontra-se agora em etapa de finalização, já os circuitos RF ainda serão projetados.

O modulador foi desenvolvido baseado em um algoritmo, usado em processamento de sinais, chamado CORDIC. Primeiramente foi feito um modelo de referência para o modulador, e após isso foi programado utilizando linguagem para hardware. Foi montado um modelo experimental para testes do modulador utilizando um kit com FPGA.

Após os testes do modulador em kits com FPGA, foi projetada uma interface serial assíncrona (UART) em linguagem VHDL para a recepção dos dados provenientes dos sensores das PCDs. Esta interface foi projetada para o padrão RS-232 e possui elementos reconfiguráveis, como a velocidade de bauds (Baud Rate) e o número de bits de dados, stop bits e bits de paridade. A interface foi prototipada e testada utilizando um kit de desenvolvimento com FPGA.

CAPÍTULO 2 – SISTEMA BRASILEIRO DE COLETA DE DADOS

2.1 – Estrutura do Sistema de Coleta de Dados.

O Sistema de Coleta de Dados [1], [2] do INPE é formado por três satélites (SCD1, SCD2 e CBERS2), várias redes de Plataformas de Coleta de Dados (PCD's) espalhadas pelo território brasileiro e ainda por duas estações de recepção em terra (Cuiabá e Alcântara) e um Centro de Missão em Cachoeira Paulista.

Os sinais gerados nos sensores das PCD's são transmitidos para os satélites, que apenas retransmitem os dados para as estações terrenas, destas os dados são encaminhados para o Centro de Missão que realiza tarefas de processamento dos dados e armazenamento além de disseminar esses dados para os usuários, que tem acesso aos dados pela internet. A figura a seguir mostra a estrutura do sistema de coleta de dados brasileiro.



Figura 1 – Sistema Brasileiro de Coleta de Dados (SBCD).

Os satélites SCD1, SCD2 e CBERS2 operam em duas faixas de frequência UHF para recepção das mensagens transmitidas pelas plataformas de Coleta de Dados: em torno de 401,62 MHz e de 401,65 MHz. Os sinais recebidos a bordo dos satélites são retransmitidos para o solo na Banda S (2.267,52 MHz) e, no caso do CBERS2 também em UHF (462,5 MHz).

Os dados processados no Centro de Missão de Cachoeira Paulista são usados pelo INPE para realizar previsões meteorológicas que são usadas pelos mais diversos meios de comunicação em massa.

2.2 – Plataformas de Coleta de Dados (PCD's).

Seguindo uma necessidade de se obter informações meteorológicas em lugares remotos do território foram criadas as Plataformas de Coleta de Dados. Antigamente a única forma de se colher dados meteorológicos era instalando aparelhos de registro e inspecionando os mesmos para verificar as medidas. Com o surgimento das PCD's, o sistema de supervisão meteorológica para futuras previsões se tornou muito mais rápido e confiável e cresce cada vez mais com a instalação de novas PCD's.

No território brasileiro estão instaladas cerca de 600 plataformas espalhadas por áreas estratégicas em todos os estados brasileiros. As PCD's podem colher dados meteorológicos, hidrometeorológicos ou ainda agrometeorológicos, além dos usos principais as plataformas estão sendo usadas hoje para outros fins como monitoramento e rastreamento de animais, controle de nível de reservatórios entre outros. De acordo com o site do INPE no total são cerca de 110 usuários dos dados colhidos pelas PCD's, entre empresas privadas, instituições de pesquisa e universidades.

As PCD's são constituídas de vários sensores, um painel solar para alimentação e um transmissor, que envia os dados para os satélites do sistema. Numa plataforma normal estão presentes cerca de 8 sensores, são eles:

- Sensor ultrasônico de vento – Através de 3 sensores mede velocidade e direção do vento.
- Sensor de temperatura e umidade relativa do ar – Mede a temperatura e umidade atmosférica.
- Sensor de radiação solar global e PAR – Mede os índices de radiação solar fotossintética ativa.
- Pluviômetro – Sensor de medida de precipitação atmosférica.
- Barômetro – Sensor de pressão atmosférica.
- Saldo radiômetro – Sensor de radiação total líquida.
- Sensor de temperatura do solo.
- Sensor de fluxo de calor no solo.

A seguir uma figura de uma PCD com seus sensores.



Figura 2 – Sensores de uma PCD.

Cada sensor é responsável por transformar cada grandeza em um sinal elétrico que será interpretado por um circuito mais adiante. Esse circuito de interpretação envia os sinais ao transmissor que processará esses sinais para que possa enviá-los aos

satélites. Nesse transmissor várias tarefas de processamento de dados, como modulação, amplificação entre outros. Um transmissor usado em PCD's no sistema brasileiro deve transmitir na frequência de 401,62 MHz, que é o padrão dos satélites do sistema.

Este projeto tem como objetivo implementar uma nova solução de transmissor para PCD's, utilizando circuitos reconfiguráveis que darão mais flexibilidade aos transmissores.

2.3 – Transmissores e técnicas de modulação usadas.

Como já dito anteriormente, um transmissor de PCD do sistema brasileiro deve transmitir à frequência UHF de 401,62 MHz, e deve obedecer também aos parâmetros do sistema francês ARGOS. Cada transmissor deve ser homologado pelo sistema antes de ser usado de fato.

Os transmissores possuem um modulador e módulos RF. A modulação usada é PSK (*Phase Shift Key*), com índice de modulação $60^\circ/V$. A modulação PSK modifica a fase do sinal da portadora de acordo com o sinal informação. Neste caso, como o sinal informação é binário a modulação só modifica a portadora em dois ângulos: -60° e 60° . Na figura a seguir é mostrado um exemplo onde o sinal informação está em azul, a portadora em verde e o sinal modulado em vermelho.

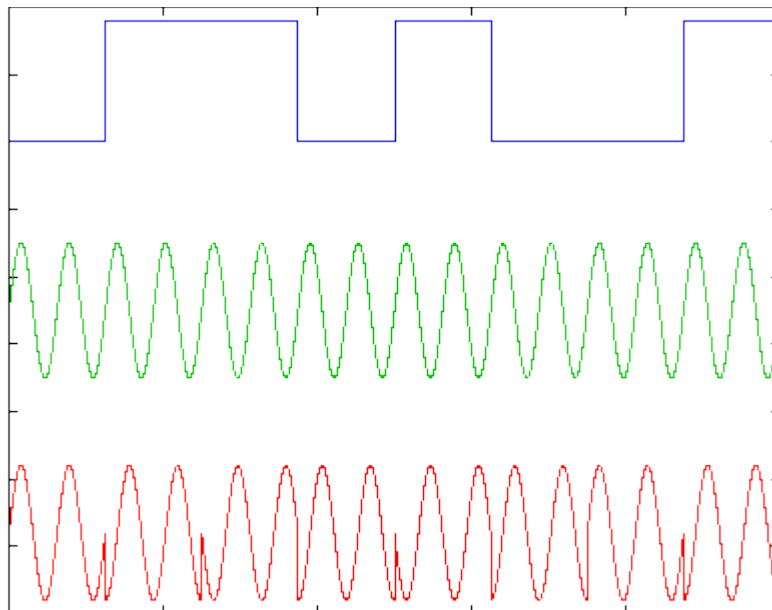


Figura 3 – Modulação PSK.

O projeto em andamento implementa um transmissor que deverá possuir tal configuração. O modulador já foi implementado utilizando a linguagem VHDL e baseado em um algoritmo, muito utilizado em processamento de sinais, chamado CORDIC. Os módulos RF ainda serão projetados.

CAPÍTULO 3 – CORDIC (ALGORITMO PARA PROCESSAMENTO DE SINAIS)

3.1 – Estrutura do algoritmo CORDIC.

O modulador desenvolvido neste projeto foi baseado em um algoritmo, largamente usado em processamento de sinais, chamado CORDIC. O CORDIC, sigla para *Coordinate Rotation Digital Computer*, é um algoritmo iterativo que realiza a rotação de um vetor de coordenadas utilizando para isso apenas operações de soma/subtração e deslocamento (multiplicação por potências de 2). Através do uso do CORDIC é possível realizar diversas tarefas de processamento digital de sinais e também de SDR (*Software Defined Radio*). [3], [4]

O módulo CORDIC possui dois modos de operação: modo rotacional (*Rotation Mode*) e modo vetorial (*Vectoring Mode*). Ambos os módulos possuem 3 entradas (X_0 , Y_0 e Z_0) e 3 saídas (X_n, Y_n e Z_n). As figuras a seguir mostram o módulo de cada modo de operação com suas saídas e entradas.

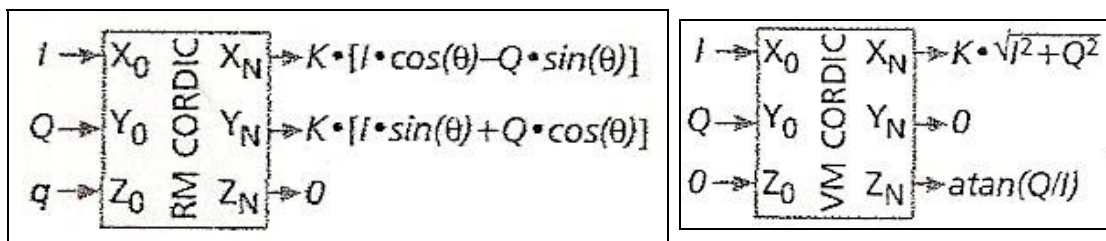


Figura 4 a – CORDIC Rotation Mode

b – CORDIC Vectoring Mode

Como pode-se observar nas saídas dos módulos, o modo rotacional realiza a rotação de um vetor (I, Q) em um ângulo q , já o modo vetorial realiza a conversão cartesiano-polar. Observa-se também que em ambos os módulos o vetor resultante sofre uma multiplicação por uma constante K .

Apenas observando os modos de operação do CORDIC já se pode concluir que para a obtenção de um modulador usa-se o modo rotacional, pois na saída apresenta sinais senoidais assim como moduladores.

Para obter tais funções de saída o CORDIC Rotation realiza as suas iterações obedecendo as seguintes equações:

$$\begin{aligned} X_{i+1} &= X_i - d_i \cdot 2^{-i} \cdot Y_i, \\ Y_{i+1} &= Y_i + d_i \cdot 2^{-i} \cdot X_i, \\ \theta_{i+1} &= \theta_i - d_i \cdot \arctg(2^{-i}); \end{aligned}$$

Onde a direção de rotação é dada por:

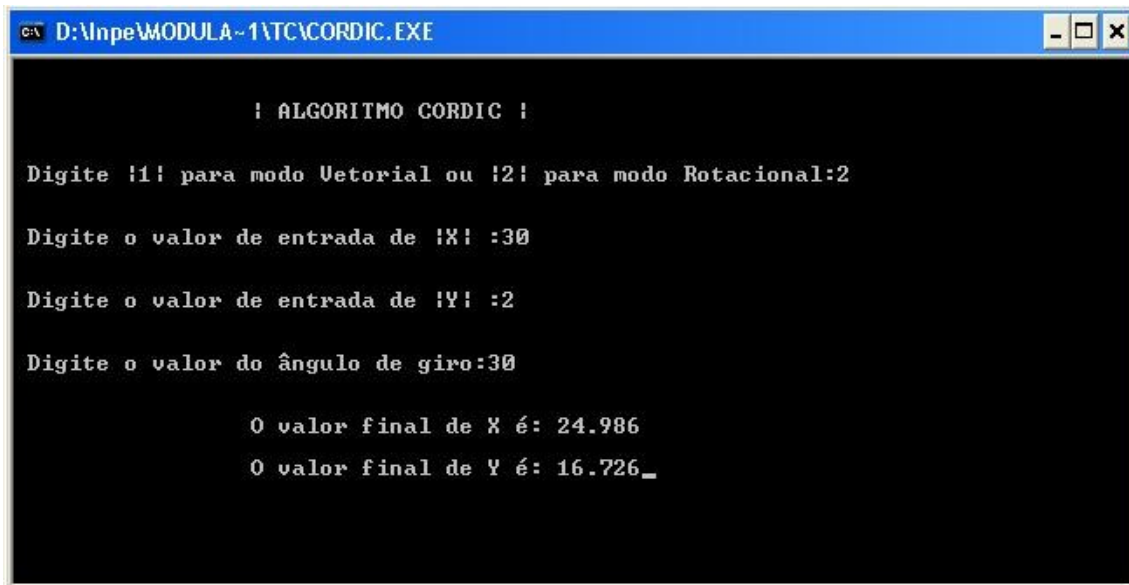
$$d_i = \text{sign}(\theta_i).$$

Os valores de \tan^{-1} devem ser colocados numa memória para que o algoritmo busque a cada iteração.

3.2 – Modelo de referência em linguagem C.

Primeiramente foi criado, utilizando a linguagem C, um modelo de referência para o módulo CORDIC. Esse modelo de referência será utilizado para verificação do

módulo feito em linguagem de descrição de hardware. Foi desenvolvido um pequeno programa para testar o modelo de referência do CORDIC. A figura a seguir mostra a execução desse programa.



```
! ALGORITMO CORDIC !

Digite !1! para modo Vetorial ou !2! para modo Rotacional:2

Digite o valor de entrada de !X! :30

Digite o valor de entrada de !Y! :2

Digite o valor do ângulo de giro:30

O valor final de X é: 24.986
O valor final de Y é: 16.726_
```

Figura 5 – Programa do modelo de referência do CORDIC.

Além de servir como modelo de referência, o CORDIC programado em linguagem C, foi útil para uma compreensão mais exata sobre o algoritmo.

3.3 – Bloco Funcional CORDIC (circuito RTL).

Depois de desenvolvido o modelo de referência do modulador começaram os trabalhos em linguagem de descrição de hardware (VHDL). Primeiramente foi tentada uma programação mais comportamental do CORDIC, ou seja, abstraindo um pouco o circuito e utilizando uma sintaxe mais de alto nível. Com a presença de alguns problemas, relativos à sintaxe de VHDL, partiu-se para uma programação mais estrutural do bloco CORDIC. De acordo com as equações de iteração do CORDIC, o algoritmo foi dividido em blocos (soma/subtração e deslocamento) que foram instanciados um a um formando cada iteração até a saída.

O código do bloco CORDIC foi sendo aprimorado e a versão atual tem uma estrutura chamada *pipelined*, com registradores entre cada iteração. A figura 6 mostra um trecho do código VHDL do CORDIC. Com a estrutura *pipelined* o circuito ganha mais em velocidade de processamento pois a cada pulso de *clock* um novo valor pode ser colocado na entrada enquanto um outro estará na segunda iteração e assim sucessivamente. A estrutura *pipelined* do CORDIC é mostrada na figura 7.

O software utilizado na programação em VHDL foi o Quartus II da Altera, fabricante que também desenvolve FPGA's e kits de desenvolvimento. No ambiente Quartus II é possível gerar o código VHDL assim como sintetizar o circuito e programar o FPGA.


```

31 TYPE VECTOR_SIGNED IS ARRAY(0 to iter-1) OF SIGNED(N-1 DOWNTO 0);
32 TYPE SIGNED_VECTOR IS ARRAY(0 to iter) OF SIGNED(N-1 DOWNTO 0);
33
34 CONSTANT rom: VECTOR_SIGNED := ("001000000000",
35                                "000100110011",
36                                "000010011111",
37                                "000001001111", -- CONSTANTES (ARCTG(2**(-i)))
38                                "000000101000",
39                                "000000010100",
40                                "000000001010",
41                                "000000000101",
42                                "000000000010",
43                                "000000000001");
44
45 COMPONENT genericadder IS
46 PORT (
47     CTRL: IN STD_LOGIC;
48     A, B: IN SIGNED(N-1 DOWNTO 0); -- COMPONENTE DO SOMADOR/SUBTRADOR
49     Sum : OUT SIGNED(N-1 DOWNTO 0)
50 );
51 END COMPONENT;
52
53 -- MONTAGEM DA ESTRUTURA ITERATIVA DO CORDIC
54 signal YSH,XSH: VECTOR_SIGNED;
55 signal X,Y,Z: SIGNED_VECTOR;
56 BEGIN
57     GENX: genericadder port map(Zin(N-1),"000000000000",Yin,X(0)); ---
58     GENY: genericadder port map(Zin(N-1),"000000000000",Xin,Y(0)); ---- ITERAÇÃO INICIAL DO CORDIC
59     GENZ: genericadder port map(not(Zin(N-1)),Zin,"010000000000",Z(0)); ---
60
61     G: for i in 0 to iter-1 generate
62
63         GEN1: genericadder port map(Z(i)(N-1),X(i),YSH(i),X(i+1));
64         GEN2: genericadder port map(not(Z(i)(N-1)),Y(i),XSH(i),Y(i+1));
65         GEN3: genericadder port map(not(Z(i)(N-1)),Z(i),rom(i),Z(i+1));
66
67         YSH(i) <= shift_right(Y(i),1);
68         XSH(i) <= shift_right(X(i),1);
69
70     end generate;
71     Xout <= X(iter);
72     Yout <= Y(iter);
73     Zout <= Z(iter);
74
75 end cordic;
76

```

Figura 6 – Trecho do código VHDL do módulo CORDIC.

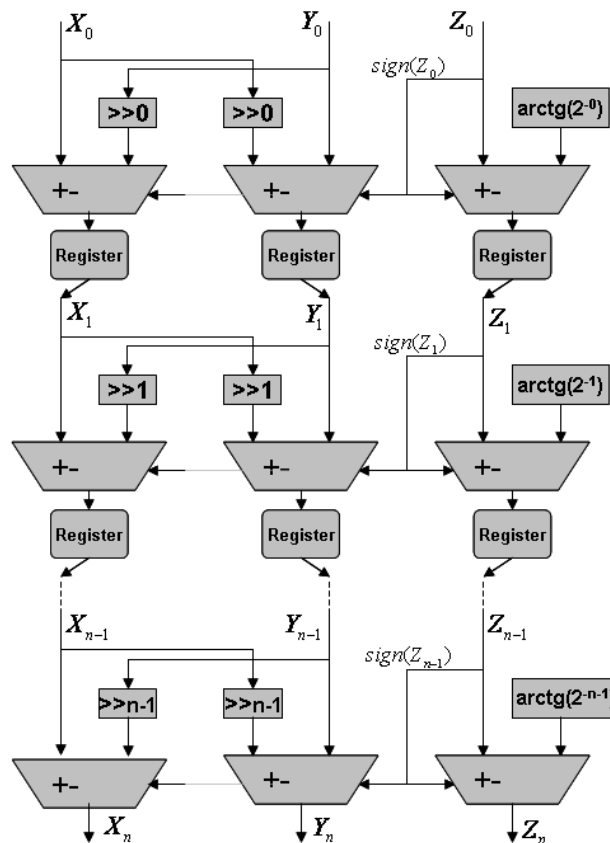


Figura 7 – Estrutura do CORDIC pipelined.

CAPÍTULO 4 – MODULADOR DIGITAL BASEADO NO CORDIC

4.1 – Esquema para modulação baseado no módulo CORDIC.

Utilizando o algoritmo CORDIC é possível implementar vários subsistemas de comunicação como: DDS (*Direct Digital Synthesizers*), moduladores analógicos (AM, FM e PM), moduladores digitais (ASK, PSK e FSK), conversores de frequência (up e down-converters), detectores de fase entre outros módulos funcionais.

Como o objetivo do projeto é implementar um transmissor, cujo modulador recebe sinal binário, foi possível utilizar o módulo CORDIC para realizar a etapa de modulação do transmissor.

Baseado em um artigo publicado na revista IEEE Communications (09/2006) [3], foi feito um esquema para modulação (figura 8) utilizando o CORDIC. Esse esquema possui um acumulador de fase e o módulo CORDIC, que realiza a conversão fase-amplitude.

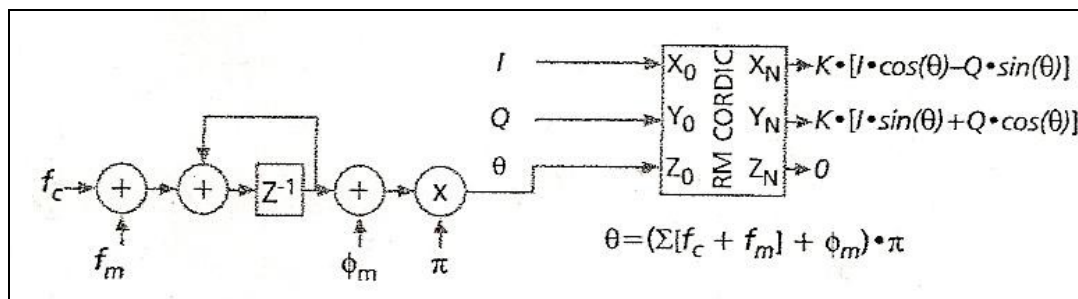


Figura 8 – Esquema para modulação utilizando CORDIC rotation.

O esquema funciona como um gerador de sinais senoidais cuja fase, frequência e amplitude podem ser modificadas através de suas entradas, em outras palavras, se trata de um modulador “genérico”. Através desse modelo é possível implementar moduladores analógicos (AM, PM e FM) e moduladores binários (ASK, PSK e FSK). Para todas as modulações a entrada f_c corresponde a frequência da portadora. Na modulação em amplitude o sinal informação (modulante) deve ser colocado na entrada I do módulo CORDIC e deve ser dividido pelo valor de K (constante de ganho do CORDIC), as demais entradas devem ser 0, assim o valor de saída do modulador será $s(n) = m(n) \times \cos(f_c \times \pi \times n)$, sendo $m(n)$ o sinal informação. Quando a modulação for na fase o sinal informação entra em ϕ_m e as demais entradas devem ser 0. Na modulação em frequência o sinal modulante entra em f_m . Podemos ver que a modulação em frequência é possível graças ao elemento de memória presente no esquema, o somador junto com o registrador faz o papel de somatório de f_m , logo se o sinal modulante entra em f_m o sinal que entra no módulo CORDIC é a integral desse sinal. Nesse caso o sinal de saída será $s(n) = \cos(f_c \times \pi \times n + (\sum m(n)) \times \pi)$.

Esse esquema de modulação foi usado como base para o projeto do modulador do transmissor, a partir dele foi feito um modelo experimental em linguagem C, e um projeto digital em VHDL. Ambos serão mostrados a seguir.

4.2 – Modelo de Referência para o modulador (Linguagem C).

Assim como no bloco CORDIC, antes do projeto ser executado em linguagem para hardware, foi feito um modelo de referência em linguagem de alto nível (linguagem C). Os resultados obtidos na saída desse modelo serão comparados com as saídas do projeto VHDL, para as mesmas entradas, com efeito de verificar o projeto digital.

Nesta etapa foi usado o sistema operacional Linux Slackware 10 e alguns aplicativos, como o compilador gcc e editores de texto. Foi feito um mini-aplicativo onde o sinal de entrada do modulador era representado por um arquivo texto (.txt) com os pontos que representavam o sinal de entrada no tempo. O programa lê o arquivo texto linha a linha e processa os dados de acordo com o esquema de modulação já apresentado, gerando, na saída, um outro arquivo texto com os pontos do sinal modulado. A figura a seguir mostra o fluxograma do modelo de referência.

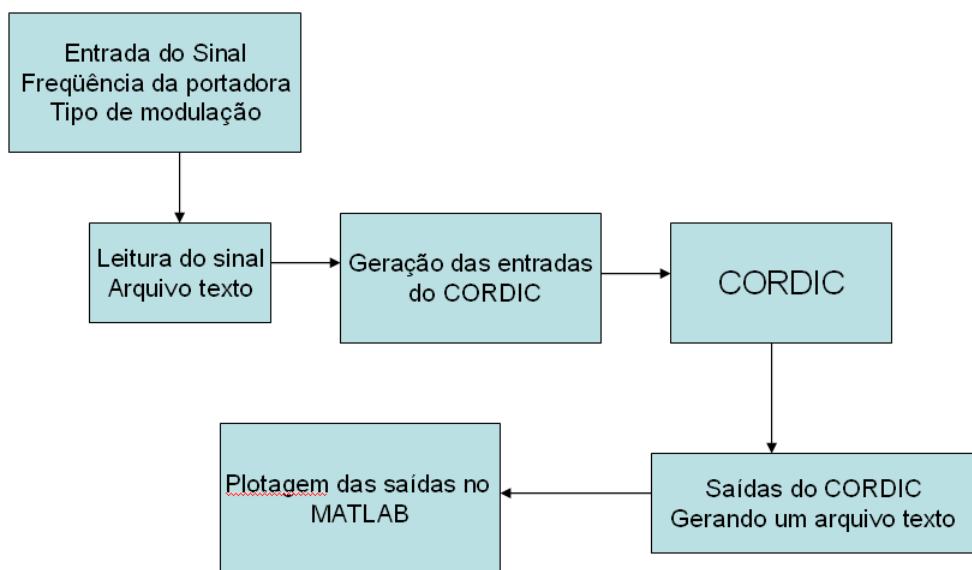


Figura 9 – Fluxo do modelo de referência para modulador.

O arquivo texto de saída pode ser lido por algum software que imprima o sinal equivalente ao sinal modulado. Neste caso foi usado o software MATLAB para verificar os sinais de saída do modelo. A figura 10 apresenta a saída do programa, onde a modulação escolhida foi AM e um sinal aleatório foi colocado na entrada.

Existem algumas metodologias para verificação de sistemas digitais, várias se utilizam de um modelo de referência, que se admite ser correto (atender as especificações do sistema). Uma metodologia bem interessante é utilizada no projeto Brazil-IP, consorcio entre universidades brasileiras para projetos de sistemas digitais, e poderá ser usada para verificação do IP Modulador.

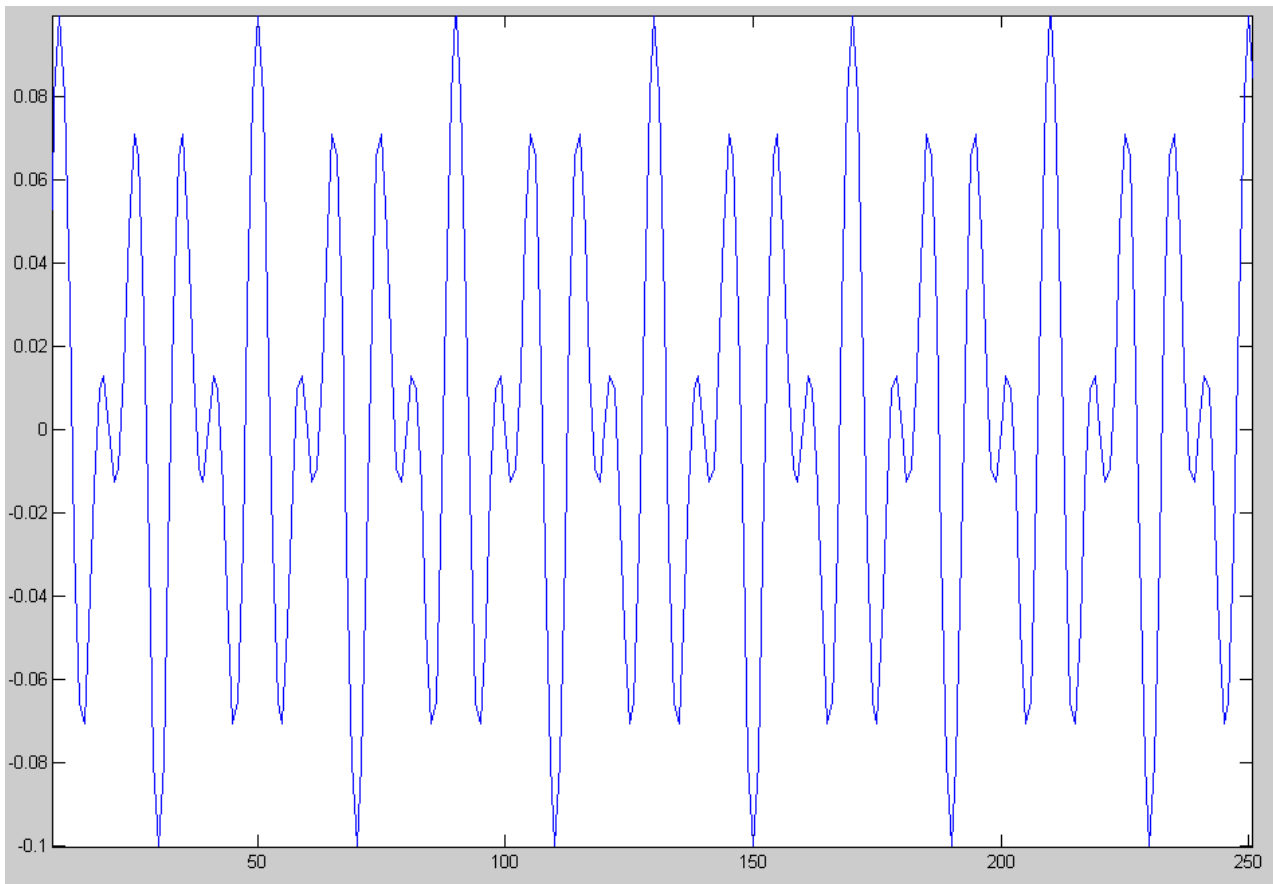


Figura 10 – Saída do modelo de referência do modulador quando escolhida modulação AM.

4.3 – IP Core Modulador (Circuito RTL).

Um IP (Propriedade intelectual) se trata de um projeto desenvolvido em linguagem para hardware que pode ser usado para prototipagem em FPGA ou ainda fabricação de chip. O IP em questão se trata de um modulador digital reconfigurável baseado no algoritmo CORDIC, na aplicação do transmissor apenas um tipo de modulação será usada (PSK), mas o IP poderá ser utilizado em outras aplicações com outros tipos de modulação ou frequências diferentes.

O início do projeto em VHDL se deu a partir do módulo funcional CORDIC, logo depois foram projetados os outros módulos do esquema apresentado no item 4.1, o registrador e o somador.

Todo o projeto desde o início foi feito de maneira a tornar o IP totalmente configurável via software. Primeiramente, todos os módulos foram feitos com uma variável do tipo GENERIC [6] para a escolha do número de bits dos dados a serem processados e uma outra para escolha do número de iterações do CORDIC, além disso, por si, o esquema do modulador já é configurável pois admite a escolha da frequência de portadora e do tipo de modulação. Além disso foi adicionado um bloco que permite a escolha de um índice de modulação, tornando o modulador o mais genérico possível.

A versão final do modulador reconfigurável possui os seguintes blocos: CORDIC, acumulador [5], MUX para escolha de tipo e índice de modulação.

A seguir a figura 11 mostra os blocos e a ligação entre eles formando o IP Modulador.

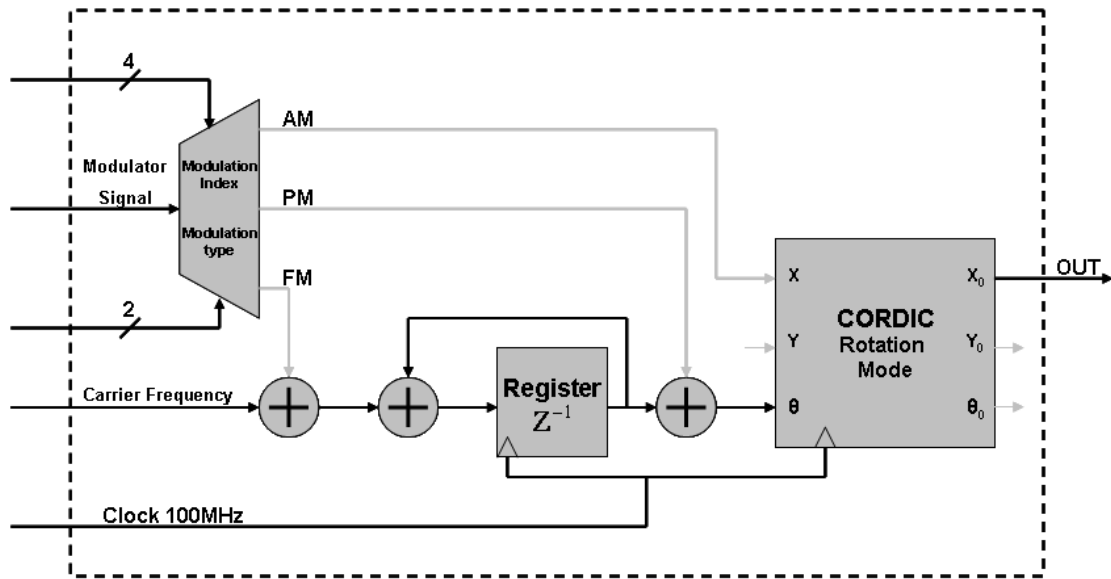


Figura 11 – Estrutura de blocos do Modulador.

Um número de 4 bits é a chave para a escolha do índice de modulação, que é na verdade uma constante (entre 1 e 16) pelo qual é dividido o sinal informação antes de entrar no modulador. Para a escolha do tipo de modulação, a entrada de seleção é um número de 2 bits que escolhe entre: modulação em amplitude, frequência, fase e portadora sem modulação. A figura 12 mostra como ficou organizado o IP na visualização RTL do ambiente Quartus II.

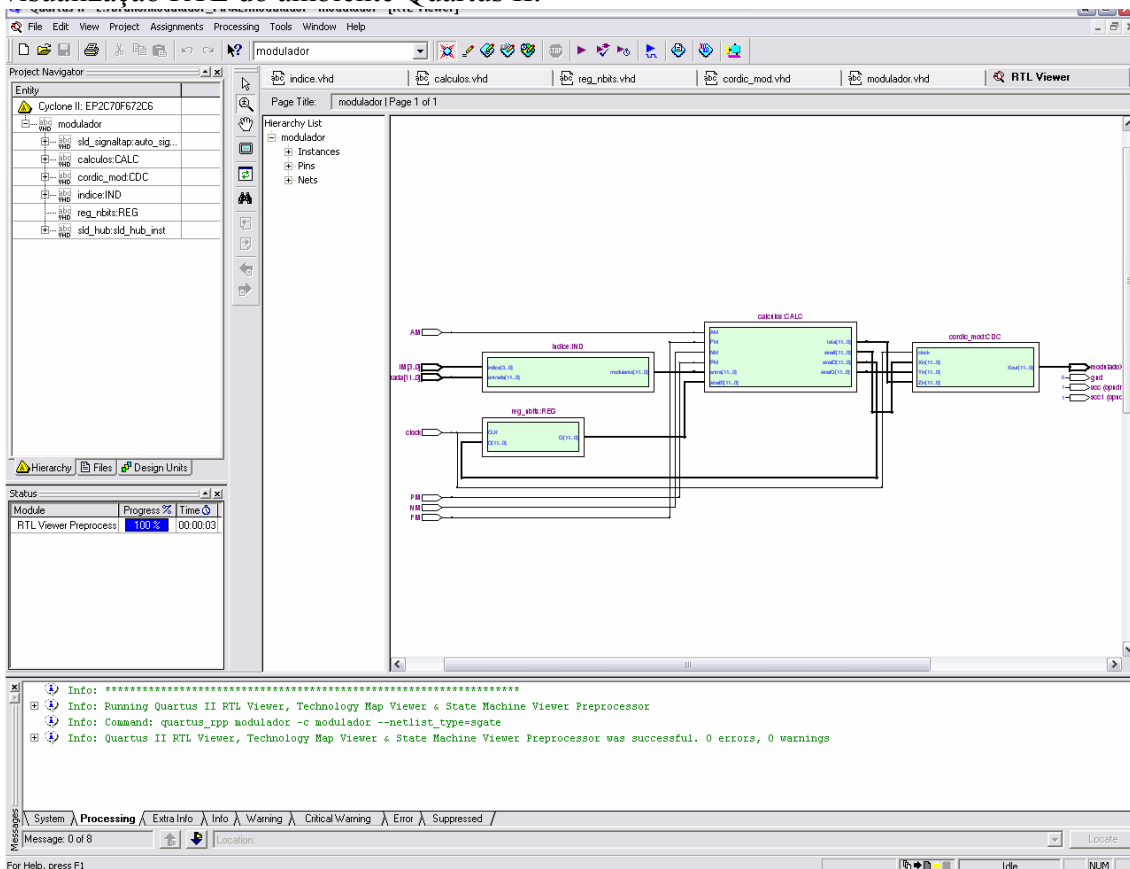


Figura 12 – Visualização nível RTL dos blocos do modulador no Quartus II.

4.4 – Modelo Experimental (Prototipação em FPGA)

Com o intuito de testar o IP Modulador e se familiarizar com técnicas de prototipagem foi proposta a montagem de um modelo experimental do modulador. O primeiro passo do modelo experimental foi a prototipagem do IP em FPGA, foi utilizado um Kit de desenvolvimento chamado Cyclone II da Altera, que possui o FPGA EP2C35F672C6ES também da Altera. Este kit possui várias interfaces de comunicação, assim como conversores A/D e D/A, botões e dip-switchs.

Para a entrada do sinal informação foi usado um conector SMA, para a escolha do tipo e índice de modulação foram usadas dip-switchs, para saída do modulador foi usado também um conector SMA. Além dessas interfaces foram usados os conversores A/D e D/A. Como qualquer projeto digital que possui elementos de memória, este também necessita de um *clock* para sincronização. O kit também possui um gerador de *clock* próprio, a frequência de *clock* usada foi de 100 MHz. A figura 13 mostra o modelo experimental com a placa, gerador de sinais e analisador de espectro.

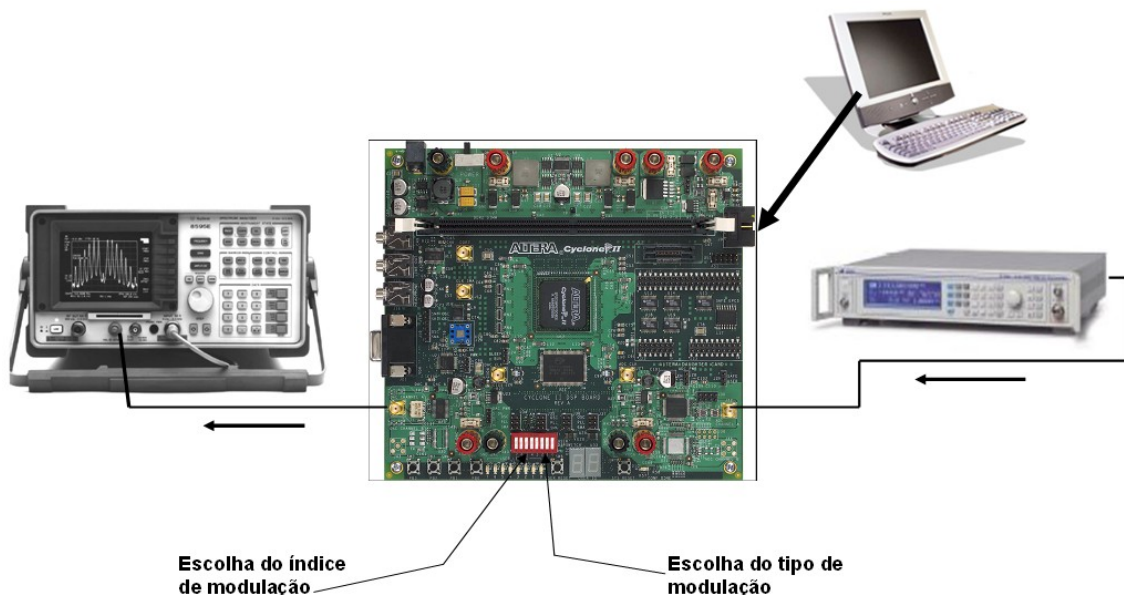


Figura 13 – Modelo Experimental do Modulador

Para programar o projeto no FPGA também foi usado o ambiente Quartus II, onde é preciso configurar os pinos que serão usados de acordo com o manual do kit de desenvolvimento.

Com esse modelo experimental foi possível comprovar a eficácia do modulador verificando o espectro (domínio da frequência) dos sinais de saída. A verificação do IP através do modelo de referência ainda se faz necessária, pois só através de análises de espectro não se garante que as saídas estão de acordo com as entradas. Alguns resultados obtidos em simulação e no modelo experimental serão mostrados adiante.

CAPÍTULO 5 – RESULTADOS E ANÁLISES

5.1 – Resultados de simulação

Para a simulação do IP Modulador, foi usada uma ferramenta do ambiente MATLAB, chamada DSP Builder, onde através do código VHDL do projeto é possível criar um bloco funcional e inserir sinais verificando as saídas correspondentes. A figura a seguir mostra o bloco funcional para o modulador gerado nesta ferramenta.

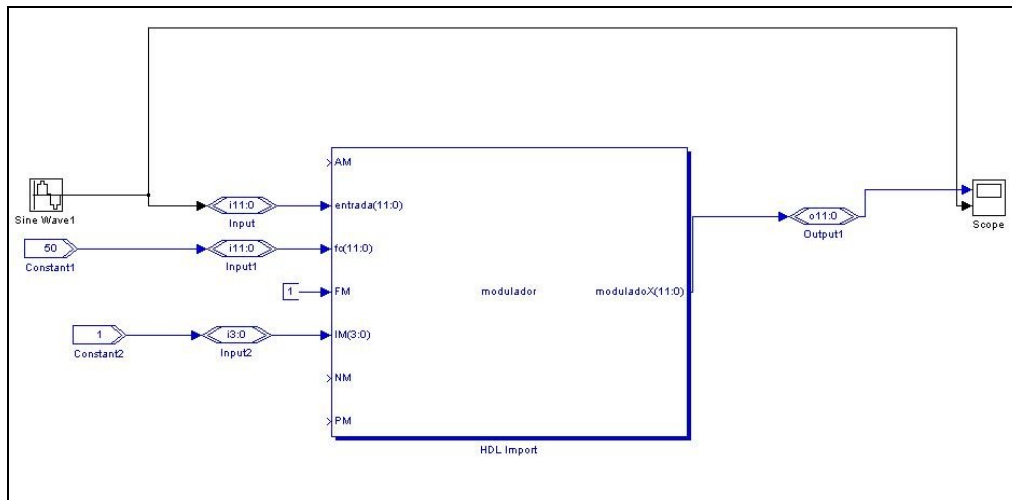


Figura 14 – Bloco funcional gerado a partir do código VHDL.

Várias simulações foram feitas usando essa ferramenta, a figura 15 mostra o resultado de uma simulação onde a entrada é uma onda quadrada, que simula um sinal binário de um sensor e a modulação escolhida é PSK. A figura 16 mostra a saída de uma modulação AM com o sinal modulante representado no quadro de baixo.

5.2 – Resultados experimentais do modulador

As figuras 17 e 18 mostram os sinais verificados no analisador de espectro utilizando o modelo experimental. Na figura 17, o primeiro sinal corresponde ao índice de modulação '0000' e o segundo sinal ao índice '1111'. Ambos os resultados foram obtidos utilizando portadora com 100 MHz e sinal banda-base de 100 KHz.

A verificação funcional do modulador será feita em paralelo aos próximos passos do projeto, utilizando ferramentas específicas no ambiente Linux.

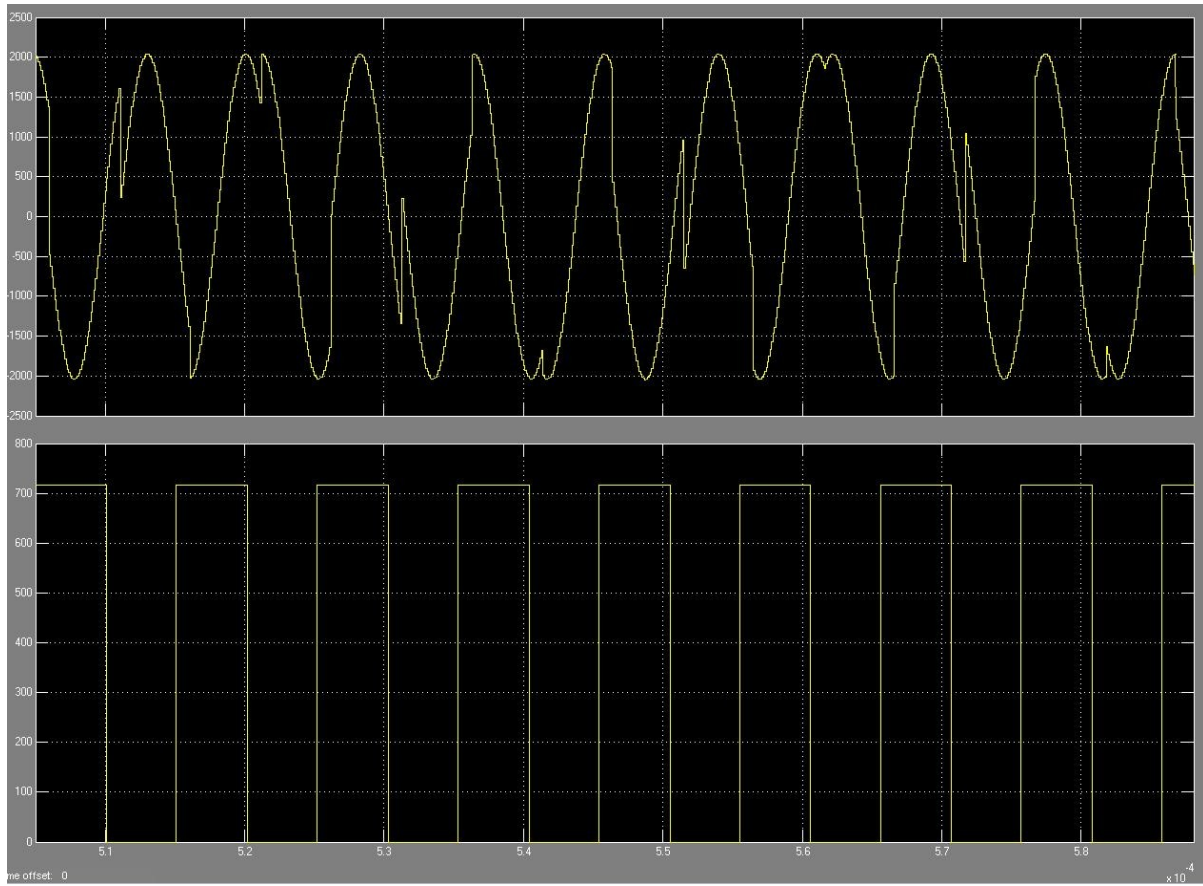
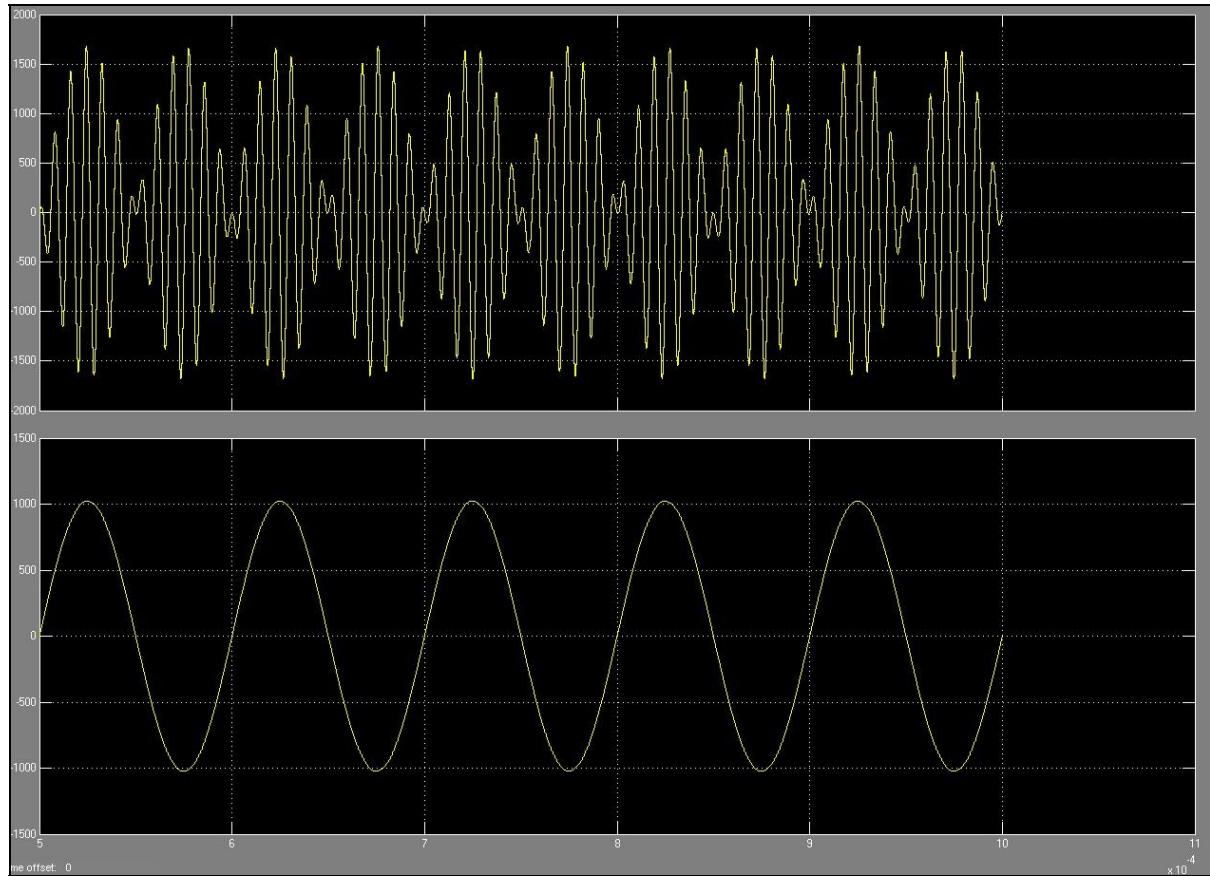


Figura 15 – Simulação de modulação PSK.



CAPÍTULO 6 – COMUNICAÇÃO SERIAL (RS-232) EM VHDL

6.1 – Interface Serial Assíncrona (UART)

Após a verificação funcional do projeto do modulador, iniciou-se o projeto de uma interface serial necessária para a recepção dos dados provenientes dos sensores, que é feita de forma serial no padrão RS-232.

Para a implementação dessa interface foi realizado em linguagem de descrição de hardware o projeto de uma UART (Universal Asynchronous Receiver/Transmitter).

A UART é um tipo de transmissor/receptor assíncrono muito utilizado em comunicação serial, geralmente é utilizado em conjunto com o padrão RS-232. A UART, na transmissão, converte um byte de dados, em um *stream* de bits correspondente, adicionando bits de parada (stop bits) e bits de paridade. Na recepção de dados, a UART recebe o *stream* de bits e extrai apenas o conteúdo de dados (bits de dados).

No caso desse projeto, foi projetada a parte de recepção do UART, onde, ao receber um *stream* de bits, são extraídos os bits de dados (correspondentes aos dados dos sensores). Na saída do receptor UART esses bits estão na forma paralela, ou seja, o receptor é um conversor serial-paralelo.

6.2 – Projeto de UART em VHDL.

A interface serial foi projetada, assim como o modulador, em VHDL, para facilitar a integração dos dois sistemas. Além do receptor UART, foram necessários outros pequenos projetos para integrar o modulador à interface serial.

Um desses projetos foi um conversor paralelo-serial que ficará entre o receptor UART e a entrada do modulador. O segundo foi um contador, utilizado para adaptar a frequência de clock do sistema para o receptor UART.

O receptor, assim como o modulador, possui elementos reconfiguráveis. Nesse caso o Baud Rate é configurável entre 4 valores (1200, 2400, 4800 e 9600), essa seleção é feita através de uma entrada de 2 bits.

A figura a seguir mostra todo o sistema digital do modulador com a interface serial, divididos em blocos.

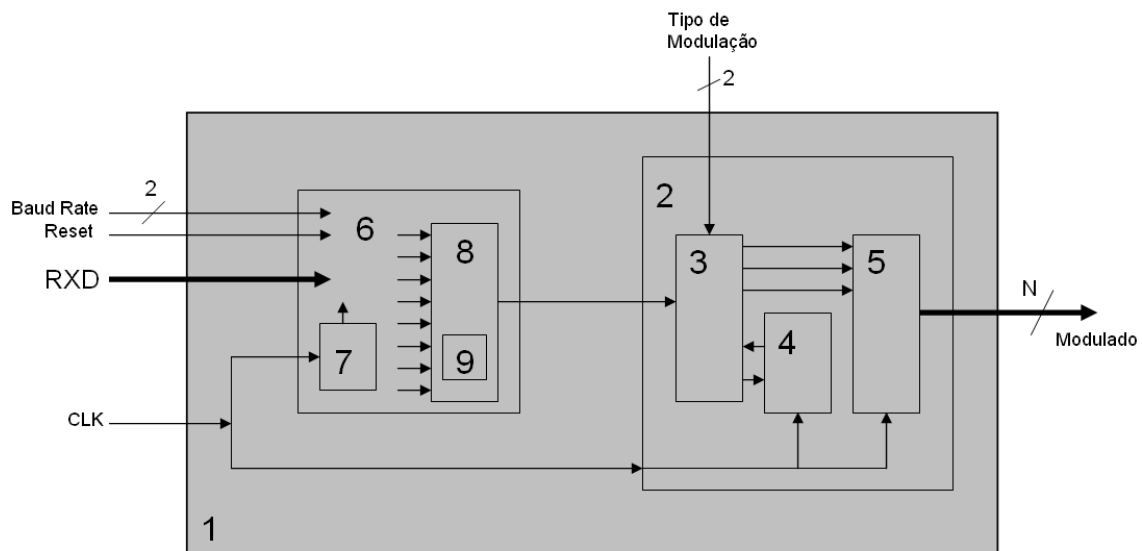


Figura 19 – Sistema Digital do Modulador com Receptor UART

1 – Sistema Completo do Modulador com interface serial.

2 – Sistema do modulador (ASK, FSK ou PSK) 60° baseado no algoritmo CORDIC.

3 – Sistemas de escolha de modulação e acumulador de fase do modulador.

4 – Registrador de N-bits do acumulador de fase.

5 – Algoritmo CORDIC.

6 – Sistema do receptor UART com Baud Rate configurável.

7 – Contador para divisão do clock em 10X.

8 – Conversor paralelo-serial para os bits de dados recebidos pela UART.

9 – Unidade de Controle para o conversor paralelo-serial.

A figura a seguir mostra a visualização do RTL do sistema completo, gerado pelo software Quartus II da Altera.

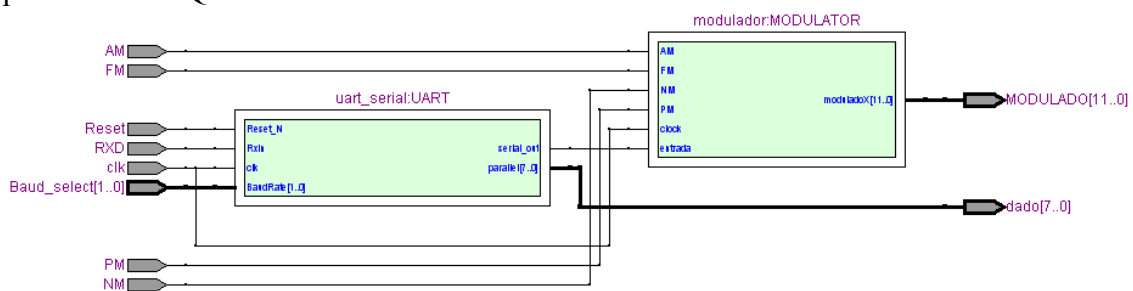


Figura 20 – RTL do Sistema Digital do Modulador com interface serial.

6.1 – Modelo de testes da interface serial.

Para testar o receptor UART projetado, foi utilizado o kit de desenvolvimento UP3-1C12, que possui o FPGA Cyclone EP1C12Q240, da Altera. Esse kit foi escolhido por ter uma entrada serial no padrão RS-232 e por ele já possuir um conversor RS-232/TTL, que é necessário para converter dos níveis de tensão utilizados no padrão RS-232 para os níveis de tensão utilizados pelo FPGA e vice-versa. O conversor presente no kit de desenvolvimento é o MAX3243.

O esquema de testes consistiu em utilizar o Hyper Terminal, acessório de comunicação do Windows, para transmitir dados, via porta serial, para o kit de desenvolvimento. Esses dados entram no receptor UART que extrai apenas os bits de dados (8 bits) e mostra esses bits nos LEDs da placa. A figura 21 ilustra o esquema de testes.

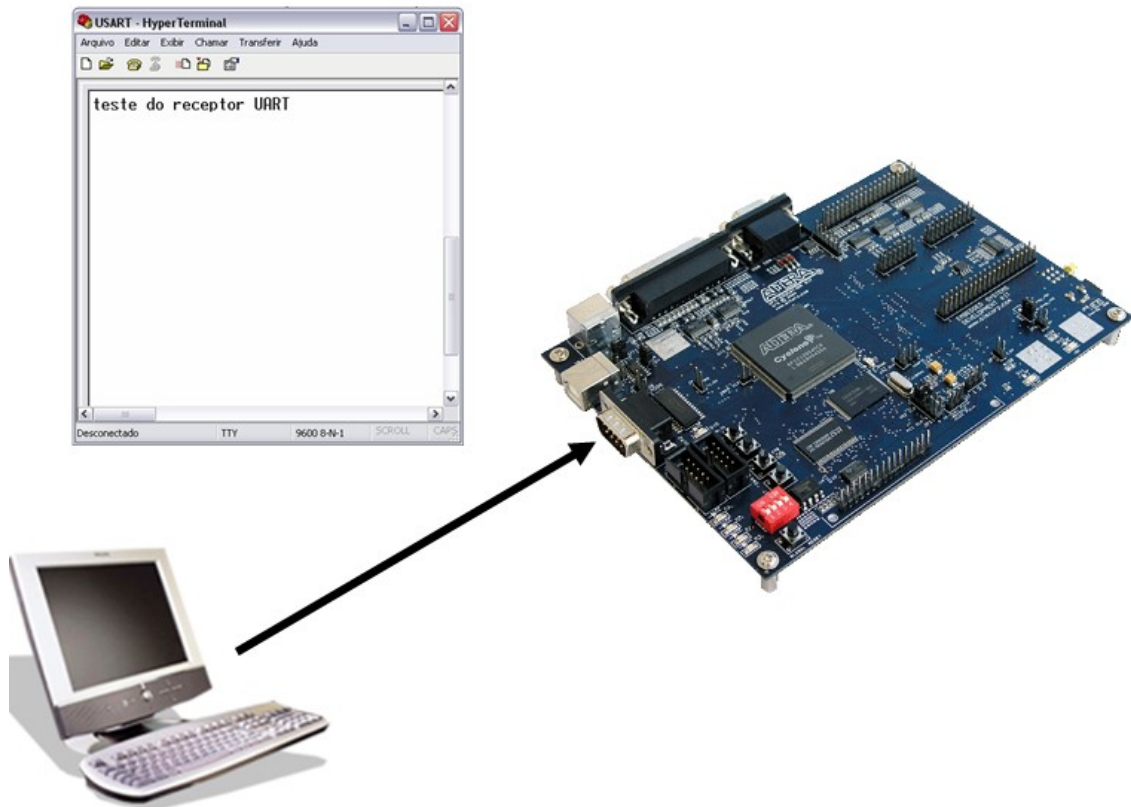


Figura 21 – Modelo de testes do receptor UART desenvolvido em VHDL.

CAPÍTULO 7 – CONCLUSÕES E TRABALHOS FUTUROS

O algoritmo CORDIC é uma poderosa ferramenta para projeto de circuitos digitais, tendo este projeto grande vantagem, por ser totalmente adaptável a vários sistemas de comunicação. Este modulador é um bloco funcional que pode ser usado como parte de outros projetos além do transmissor de PCD's.

Através desse projeto foi possível adquirir conhecimento em vários âmbitos, um artigo científico sobre o modulador foi publicado e apresentado em congresso internacional (SFORUM2007), no Rio de Janeiro e aceito em outro congresso internacional (SPLCONF 2008), em Bariloche, Argentina.

Os próximos passos do projeto são:

- Projeto de uma placa com FPGA que servirá de protótipo para o modulador.
- Testes do modulador com dados reais de PCDs.
- Acoplamento de módulo de Rádio Frequência ao modulador digital para adaptar o sinal ao padrão do Sistema de Coleta de Dados.

REFERÊNCIAS

- [1] Site do INPE: www.inpe.br Último acesso: 09 de julho de 2008.
- [2] E.A.P. Tude, C.A.I. Miranda, L.E.M. Parada, M.H.M. Costa, S.P. Pereira and V.M. de Medeiros, “Análise do Sistema de Coleta de Dados da MECB/SS”, INPE-3820-NTE/253, pp. 1-6, March 1986.
- [3] J.Valls, T. Sansaloni, A. Perez-Pascual, V. Torres, V. Almenar, “The use of CORDIC in Software Defined Radios”, IEEE Communications Magazine, September 2006.
- [4] R. Andraka, “A survey of CORDIC algorithms for FPGA based computers”, pp. 1-5, 1998.
- [5] D. De Caro, N. Petra, A.G.M. Strollo, “A 380 MHz Direct Digital Synthesizer/Mixer With Hybrid CORDIC Architecture In 0,25 mm CMOS ”, IEEE Journal Of Solid-State Circuits, vol. 42, N° 1, January 2007 pp. 1-5, 1998.
- [6] V.A. Pedroni, “Circuit Design with VHDL”, MIT Press, 2004.
- [7] Bertrand Cuzeau, “VHDL – Practical Example – Designing an UART”, ALSE, sept 2001.
- [8] UP3-1C12 Education kit - Reference Manual.
- [9] Cyclone II EP2C35 DSP Development Board - Reference Manual.
- [9] Datasheet – MAX232