

Plataforma para processamento de sinais em FPGA (PS-FPGA)

Bruno Marcos Espindola

Aluno, Tecnólogo em Sistemas de Telecomunicações, Instituto Federal de Santa Catarina
brunomarcosespindola@gmail.com

Glaucio Bertelli Peres

Aluno, Tecnólogo em Sistemas de Telecomunicações, Instituto Federal de Santa Catarina
glaucio1986@gmail.com

Marcos Moecke

Prof., Sistemas de Telecomunicações, Instituto Federal de Santa Catarina
moecke@ifsc.edu.br

Resumo- Os FPGAs são dispositivos programáveis capazes de implementar circuitos digitais. Neste artigo são apresentados módulos para aquisição, reprodução, filtragem, demodulação e armazenamento de sinais realizados utilizando um kit com FPGA da Altera. Também foram implementados alguns cenários especiais utilizando os blocos criados em VHDL e blocos IP já prontos, de forma a exemplificar e incentivar a programação nestes dispositivos.

Palavras-chave: FPGA. Processamento de Sinais. VHDL. Gerador Senoidal. Filtros Digitais

Abstract- FPGAs are programmable devices that are capable of implementing digital circuits. In this article, signal acquisition, playback, filtering, demodulation and storage are shown using an Altera's FPGA development kit. In order to exemplify and encourage use of the built-in blocks and those implemented in VHDL some special scenarios were implemented using these block.

Keywords: FPGA. Signal Processing. VHDL. Sinusoidal Generator. Digital Filter.

1 Introdução

O projeto de equipamentos de telecomunicações nos últimos anos passou por muitas inovações, principalmente com a evolução dos circuitos de lógica programável. Dentre as possíveis soluções para projeto, a escolha da melhor alternativa é condicionada por fatores como escala de produção, tempo e custo de desenvolvimento, treinamento da equipe de desenvolvedores, e também a capacidade de processamento necessária. No mercado global atualmente existem diversas alternativas, dentre elas, os circuitos FPGAs (*Field Programmable Gate Arrays*), que apresentam uma grande capacidade de processamento em função do paralelismo das operações. Devido à grande flexibilidade de programação e campos de uso, esses dispositivos são uma boa escolha para a implementação de sistemas (PEDRONI, 2007).

Este trabalho implementa um sistema para a aquisição, processamento e reprodução de som e voz, utilizando ferramenta de desenvolvimento para FPGA. O resultado é uma plataforma para processamento de sinais em FPGA (PS-FPGA) que possui um conjunto de módulos para essas funções, e que pode ser utilizada em projetos didáticos para a implementação em tempo real de sistemas inicialmente desenvolvidos em ambiente Matlab/Simulink e que envolvam intenso processamento de sinais (ESPINDOLA B. M.; MOECKE, 2010).

A sequência deste artigo apresenta na Seção 2 uma descrição dos módulos implementados destacando alguns aspectos teóricos dos processos envolvidos. Na Seção 3 serão apresentados os resultados obtidos com a implementação, e a Seção 4 apresentará conclusões e proposta para trabalhos futuros.

2 Desenvolvimento

A PS-FPGA foi implementada usando dispositivos da ALTERA disponíveis no kit de desenvolvimento denominado *DSP Development Kit - Stratix II Edition*. Foram programados em VHDL ou usando IPs proprietários da ALTERA e estão disponíveis para uso os seguintes módulos: controle da aquisição do sinal de voz usando o conversor A/D; controle da reprodução do sinal de voz usando o conversor D/A; armazenamento em memória de janelas de tempo de sinais de voz; subamostragem de sinais; filtros com função de transferência programável; gerador de sinais senoidal programável; misturador de sinais.

No restante dessa seção são apresentadas as ferramentas utilizadas para o desenvolvimento dos módulos e posterior aplicação dos mesmo em cenários de teste, e as técnicas utilizadas no desenvolvimento projeto, bem como uma descrição dos módulos desenvolvidos.

2.1 Ferramentas e hardware usado

Para o desenvolvimento da plataforma PS-FPGA foram utilizadas diversas ferramentas da ALTERA, e uma plataforma de *hardware* desse fabricante. O *hardware* utilizado foi o kit de desenvolvimento DSP EP2S60, que é voltado ao desenvolvimento de diversos projetos e sistemas com processamento de sinais digitais. Ele é baseado em um FPGA EP2S60F1020C3 da família Stratix II ligado a dispositivos com: conversores D/A de 12 bits que suportam uma taxa de até 125MHz; conversores A/D de 14 bits que suportam uma taxa de até 165MHz; oscilador de 100MHz; mostradores de sete-segmentos; botões do tipo contato-momentâneo (*push-button*); e leds. Tanto o FPGA como os dispositivos periféricos foram utilizados para a implementação da PS-FPGA.

Para a programação do *hardware* do FPGA foi principalmente utilizado o *software* Quartus II, que é um ambiente de desenvolvimento que permite realizar a programação do FPGA utilizando uma linguagem de descrição de *hardware* (*Hardware Description Language* - HDL). O ambiente também permite compilar o programa, definir a conexão dos pinos do FPGA, controlar o uso dos recursos disponíveis na placa, realizar a síntese

do circuito no FPGA, e carregar o arquivo de configuração obtido para o FGPA via interface JTAG.

Antes do teste no *hardware* optou-se por realizar a simulação exaustiva com o Modelsim-Altera, o qual permite a criação de um banco de testes para detectar erros de implementação dos módulos e cenários sem a necessidade de realizar síntese no FPGA, tornando mais ágil e rápido o desenvolvimento dos módulos.

Alguns módulos implementados neste projeto foram desenvolvidos com módulos de Propriedade Intelectual (*Intellectual Property - IP*), que são blocos de códigos HDL com interfaces de configuração predefinidas pelo fabricante do IP. Alguns IPs são liberados para uso acadêmico, e por isso simplificam a implementação de projetos, reduzindo o tempo de desenvolvimento e testes. Outros desenvolvedores disponibilizam em parceria com os fabricantes de FPGAs estes módulos IPs vendendo as licenças de uso por implementação.

2.2 Circuitos FPGAs

Os FPGAs são circuitos integrados que possuem a capacidade de serem ajustados de forma a criar o *hardware* necessário para determinada aplicação. Estes dispositivos contêm um grande número de elementos lógicos idênticos que podem ser configurados individualmente e interconectados por uma matriz de trilhas condutoras e chaves programáveis. Além destes elementos lógicos básicos, os FPGA's podem possuir componentes mais complexos como multiplicadores dedicados, blocos de memória e circuitos PLLs, que permitem otimizar o uso dos elementos lógicos e desenvolver projetos mais eficientes. A programação de um FPGA é normalmente feita através de um arquivo binário gerado utilizando-se um ambiente próprio de desenvolvimento fornecido pelo fabricante do FPGA. Esse arquivo contém as informações necessárias para especificar as funções de cada bloco lógico e realizar as interconexões entre eles.

As linguagens HDLs usadas para modelar um *hardware* programável, através de um *software*, dentre elas temos: AHDL, Handel-C, SDL, ISP, ABEL, Verilog e VHDL. Entre estas, as duas últimas são as mais comumente disponibilizadas nos ambientes de desenvolvimento para plataformas FPGAs, e por isso as mais utilizadas na academia e na indústria (PEDRONI, 2007). No projeto utilizou-se a linguagem VHDL (*Very High Speed Integrated Circuit Hardware Description Language*) por mais comum na literatura e ser a linguagem utilizada em cursos de graduação. Essa linguagem permite fazer a modelagem estrutural e a modelagem comportamental de um sistema assim como programar bancos de teste para o *hardware* desenvolvido. No modelo estrutural os módulos de *hardware* são descritos a partir dos seus componentes básicos e pelas suas interconexões. No modelo comportamental os módulos são descritos a partir do resultado que se deseja, sem importar a forma como o *hardware* será criado.

2.3 Filtros

Os sistemas práticos estão sujeitos a ruídos aleatórios e perturbações que podem dificultar a análise de um sinal ou outro procedimento. No processamento de sinal, a

função de um filtro é eliminar partes indesejadas, tais como ruídos, e principalmente, extrair partes úteis fazendo o tratamento das informações no domínio da frequência, separando, classificando e medindo sinais.

A implementação em *hardware* da função de transferência de filtros digitais é normalmente feita usando elementos básicos como multiplicadores, somadores e elementos de atraso, utilizando uma das possíveis formas de realização. A implementação digital resulta em uma maior precisão, e flexibilidade comparada com filtros analógicos. Algumas das vantagens desses filtros são: facilidade de alteração da função de transferência do filtro através da mudança do valor dos seus coeficientes; possibilidade de implementação de filtros de ordens acima de 8; a não variação dos parâmetros dos filtros com a temperatura e envelhecimento; a reprodutibilidade exata do filtro em linhas de produção sem a necessidade de ajustes; precisão da resposta em frequência é definida pela forma de realização e quantidade de bits utilizados na representação dos coeficientes; maior simplicidade no projeto usando ambientes de desenvolvimento apropriados (SHENOI, 2006).

Os filtros digitais são criados utilizando elementos básicos como multiplicadores, somadores e elementos de atraso, de acordo com o algoritmo determinado pela função de transferência dos filtros e suas formas de realização, que é como a função transferência é implementada em *hardware*. Eles oferecem uma maior flexibilidade e versatilidade em relação aos filtros analógicos já que vem a ser uma tarefa fácil alterar o valor dos coeficientes de um filtro digital, mudando assim a sua função de transferência. Além disso, os filtros digitais são mais estáveis, visto que os filtros analógicos são sujeitos a variações de temperatura, variações de valores devido aos componentes usados no circuito, e outros parâmetros que dependem da aplicação e do projeto (SHENOI, 2006).

Os filtros digitais são descritos por dois tipos de funções de transferência: filtros de resposta ao impulso infinita (IIR) e filtros de resposta ao impulso finita (FIR). Os filtros FIRs diferenciam-se dos IIRs por serem não recursivos, portanto a resposta de um filtro FIR depende exclusivamente do sinal de entrada atual não importando os valores de saída anteriores.

Os filtros FIRs possuem algumas vantagens em relação aos IIRs, tais como: podem ser projetados de forma a ter fase sempre linear; são sempre estáveis; as amostras da resposta ao impulso são os mesmos valores dos coeficientes da função de transferência. Já os filtros IIRs possuem a vantagem de exigirem uma menor ordem para realizarem uma determinada resposta em frequência.

Optou-se por filtros FIR neste trabalho, por terem fase linear e disporem de IPs que permitem a sua implementação de forma simples e rápida. Para a implementação dos filtros neste projeto foi utilizada a ferramenta da Altera denominado *FIR Compiler*. Essa ferramenta pode ser chamada diretamente a partir do *Quartus II* (ambiente de desenvolvimento para FPGA) através do *MegaWizard*. O filtro obtido é transformado em

um módulo VHDL e pode ser incluído em projetos mais complexos através do editor esquemático. Usando o *FIR Compiler* é possível projetar e implementar diretamente alguns tipos de filtros FIR ou importar os coeficientes da função de transferência do filtro. Neste último caso o projeto do filtro deve ser feito usando outra ferramenta tal como o Fdatool do Matlab e os coeficientes do filtro projetado posteriormente exportados.

2.4 Modulação e demodulação

Sinais em banda base produzidos por diversas fontes de informação nem sempre estão adequados para a transmissão através de um determinado canal. Estes sinais são normalmente modificados para facilitar a transmissão. Este processo de conversão é chamado de modulação. Neste processo, o sinal em banda base é usado para modificar algum parâmetro de uma portadora de alta frequência (LATHI, 1998). Esse processo visa principalmente compartilhar o meio de transmissão entre diversas transmissões de sinais transmitindo diferentes sinais sem que ocorra sobreposição ou perda de informação. Outra aplicação é a redução do tamanho da antena necessária para a irradiação do sinal, devido ao deslocamento do espectro do sinal.

Entre as modulações, adotamos a modulação em amplitude na qual a amplitude A de uma portadora senoidal $A \cos(\omega_c t + \theta_c)$ varia de acordo com um sinal modulante em banda base $m(t)$. Este tipo de modulação desloca o espectro $M(\omega)$ do sinal $m(t)$ para a frequência da portadora

$$m(t) \times \cos(\omega_c t) \Leftrightarrow \frac{1}{2} [M(\omega + \omega_c) + M(\omega - \omega_c)] \quad (1)$$

O processo de demodulação ocorre no receptor, no qual o sinal em banda base $m(t)$ é recuperado.

2.5 Aquisição e reprodução de sinais

Para o processamento digital de um sinal analógico é necessário que uma conversão de analógico para digital e vice-versa seja feita. Os dispositivos utilizados para fazer essa conversão são: o conversor analógico para digital (A/D) e o conversor digital para analógico (D/A). No projeto desenvolvido foi utilizado o kit DSP EP2S60 da ALTERA que dispõe de dois conversores A/D AD9433 com saída de 12 bits e dois conversores D/A DAC904 com entrada de 14 bits.

Os conversores D/A são dispositivos eletrônicos que discretizam o sinal analógico, tanto na dimensão temporal (amostragem em intervalos uniformes) como na dimensão amplitude (quantização). O valor das amostras resultantes são representadas através de bits. Quanto maior a quantidade de bits, menor será o erro de quantização, pois é possível representar um maior número de níveis de referência analógicos. O sinal discreto obtido manter fielmente a informação original, isto é, sem perda. Para que isso ocorra é necessário que a taxa de amostragem f_s utilizada seja pelo menos duas vezes maior do que a frequência máxima do sinal a ser amostrado, conforme a taxa de Nyquist.

$$f_s \geq 2 f_{\max} \quad (2)$$

No processo de conversão D/A cada amostra digital é transformada em um valor correspondente de tensão ou corrente. No tempo entre as amostras o valor anterior é normalmente mantido, resultando em uma significativa distorção de alta frequência. Através de um filtro passa baixa analógico com frequência de corte de $0,5f_s$ é possível remover essas frequências e recuperar o sinal.

2.6 Armazenamento

Em diversas aplicações têm-se a necessidade de armazenar informações para serem utilizadas posteriormente. Em muitos casos, vem a ser impossível realizar um projeto sem fazer uso de memórias. Nos FPGAs é possível sintetizar vários tipos de memória, sejam elas FIFO, ROM ou RAM, dependendo a necessidade encontrada no momento. Muitos dos FPGAs já possuem uma quantidade de blocos de memória prontos que são utilizados caso alguma memória seja sintetizada. Caso não haja blocos de memória suficientes, a memória é criada utilizando os elementos lógicos disponíveis no chip.

2.7 Gerador senoidal

Para implementar um gerador senoidal em um sistema utilizando FPGA existem diversas opções. A escolha da melhor alternativa depende dos recursos disponíveis no *chip* e qualidade da onda que se busca gerar. A Altera possui um IP denominado NCO *MegaCore* (ALTERA,2011b), onde é possível implementar geradores mais complexos. Este IP é um Oscilador Controlado Numericamente (*Numerically-controlled oscillator - NCO*) que possui uma maior precisão possibilitando a seleção através de uma entrada numérica de uma grande faixa de valores de frequência de saída. A sua arquitetura pode ser implementada usando o algoritmo CORDIC, estrutura ROM grande, ROM pequena e ser baseada em multiplicadores. No gerador senoidal implementado foi escolhido o algoritmo CORDIC, por apresentar uma qualidade de sinal equivalente, com um menor consumo de *hardware*.

O algoritmo CORDIC fornece uma solução de alta performance para osciladores com precisão muito alta em sistemas com pouco uso memória interna, calculando os valores das amostras do seno e cosseno. O algoritmo é baseado no conceito de rotação do fasor complexo por multiplicação de ângulo de fase por constantes sucessivamente menores. Em *hardware* digital, a multiplicação é feita apenas por expoente dois, podendo se implementado apenas com uma série de deslocamento, soma/subtração de números binários. O cálculo do seno e do cosseno de um valor de fase de entrada de é feito de forma interativa deslocando o ângulo de fase para aproximar os valores da coordenada cartesiana com o ângulo de entrada. Ao final da interação o cosseno e seno são respectivamente a coordenada X e Y de um determinado ângulo conforme mostra a Fig 1.

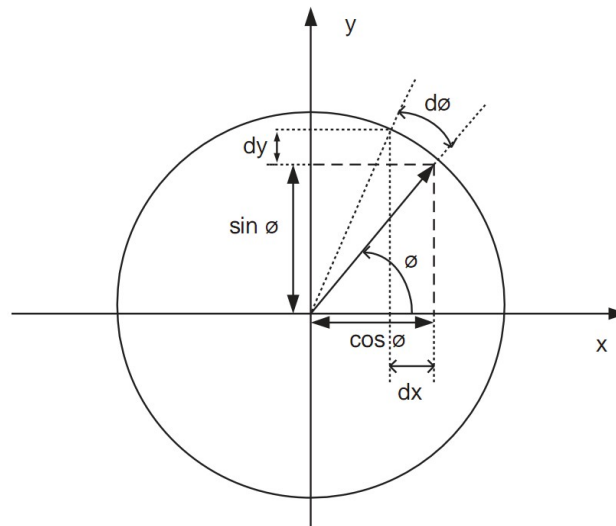


Figura 1 - Rotação CORDIC para o calculo de seno e cosseno.

No IP NCO *Megacore* existem duas opções de arquitetura para o algoritmo CORDIC: paralela ou serial. Na arquitetura em paralelo é usado um oscilador de alta performance e precisão implementado apenas com elementos lógicos, de modo que a cada ciclo de relógio um valor é disponibilizado na saída. A arquitetura em serial usa menos recursos de *hardware*, porém seu desempenho é reduzido por um fator igual ao valor de precisão de magnitude. Por exemplo, se o valor selecionado na precisão de magnitude for N , a taxa de saída e a frequência máxima do sinal é reduzida por um fator N . A implementação serial o resultado dos somadores é armazenado sendo disponibilizado uma amostra na saída a cada N ciclos de relógio. A frequência senoidal máxima que o NCO pode gerar é limitada a metade da frequência do relógio de entrada do NCO de acordo com o critério de *Nyquist*. O controle da frequência de saída f_o gerada pelo NCO depende do valor do incremento de fase ϕ_{inc} , sendo obtida por

$$f_o = \frac{\theta_{inc} \cdot f_{clk}}{2^M}, \quad (3)$$

onde M é o parâmetro *phase accumulator precision* e f_{clk} é a frequência de relógio de entrada no NCO.

3 Resultados

Com o objetivo de testar os módulos desenvolvidos foram utilizados diversos cenários que exploram características de aplicações reais da área de telecomunicações. Os cenários foram inicialmente simulados usando o *Modelsim* e em seguida sintetizados no FPGA usando o kit *DSP EP2S60*. Para efetuar as medições foram utilizados em diferentes momentos geradores externos de sinais, osciloscópios, analisadores de espectro. Em alguns cenários optou-se pela uso do analisador lógico *SignalTap*, que é integrado ao *Quartus II* e sintetizado no próprio FPGA usando a interface JTAG para a captura e apresentação em tempo real de qualquer sinal interno do FPGA.

3.1 Módulo para aquisição e reprodução de sinal

Para testar a aquisição e reprodução de sinais foi utilizado um cenário mostrado na Fig. 2. Neste caso, um gerador de senoidal gera o sinal a ser enviado para o conversor D/A, sendo o sinal analógico da saída do conversor D/A conectado a entrada do conversor A/D através de um cabo coaxial. O sinal digital do tipo *signed* de 12 bits obtido na saída do conversor A/D é convertido para *unsigned* de 14 bits pelo bloco conversor “*converter unsigned*” e convertido para analógico pelo outro conversor D/A.

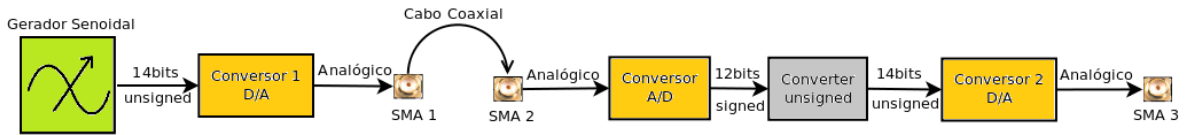


Figura 2 - Diagrama do cenário de testes de aquisição e reprodução.

Nos testes realizados, percebeu-se que frequências menores que 30 kHz sofrem uma atenuação muito grande, tornando inviável a aquisição de sinais na faixa entre 0 e 30 kHz. O motivo dessa atenuação é a existência de transformadores de RF ADT1-1WT junto aos conectores SMA. Estes transformadores têm resposta de frequência plana na faixa entre 0,4 MHz a 800 MHz atenuando fortemente os sinais fora dessa faixa. Em função dessa característica optou-se por limitar os testes a sinais na faixa de 30 kHz a 6,25 MHz.

3.2 Circuito de anti-repique

As chaves mecânicas (*push-button*) utilizadas ao serem pressionadas produzem repiques que resultam em várias transições de sinais se não forem tratadas. Para tal foi utilizado um circuito anti-repique (*debouncing*), implementado conforme este circuito mostrado na Fig. 3. Utiliza-se um bloco contador para gerar um sinal de relógio de 10ms, o qual é usado como relógio dos *flip-flops* D que estão conectados às chaves. O tempo de 10ms foi determinado de modo a eliminar o repique mecânico.

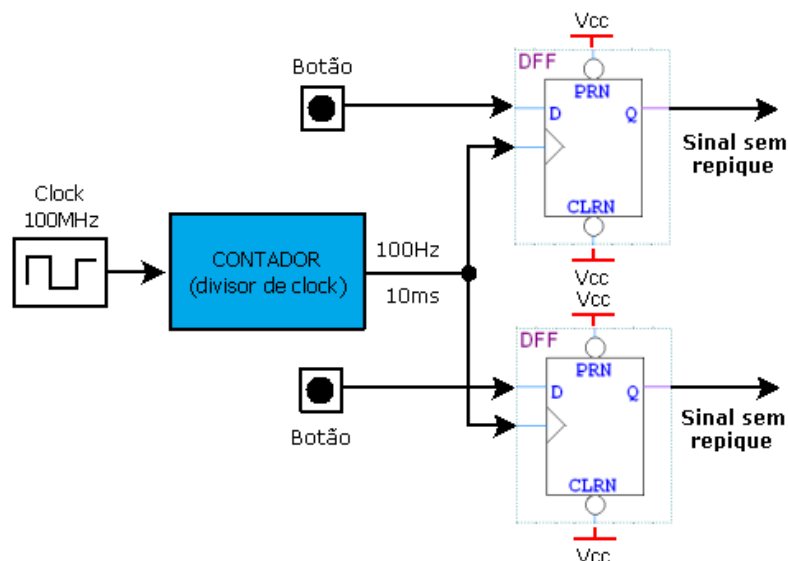


Figura 3 - Diagrama do circuito anti-repique.

3.3 Memória para armazenamento de sinal

O circuito de armazenamento de sinal foi implementado de forma que um sinal do conversor A/D ou do gerador senoidal fosse sub-amostrado a uma taxa de 44.100 amostras por segundo. Ao acionar o botão que ativa a escrita, 88.200 amostras de 12 bits correspondentes a 2 segundos de sinal são armazenados na memória para posterior processamento. Ao acionar o botão de leitura os dados armazenados são lidos e enviados para o conversor D/A. O circuito mostrado na Fig. 4 foi sintetizado usando módulos VHDL, e interligados usando o diagrama esquemático da ferramenta Quartus II.

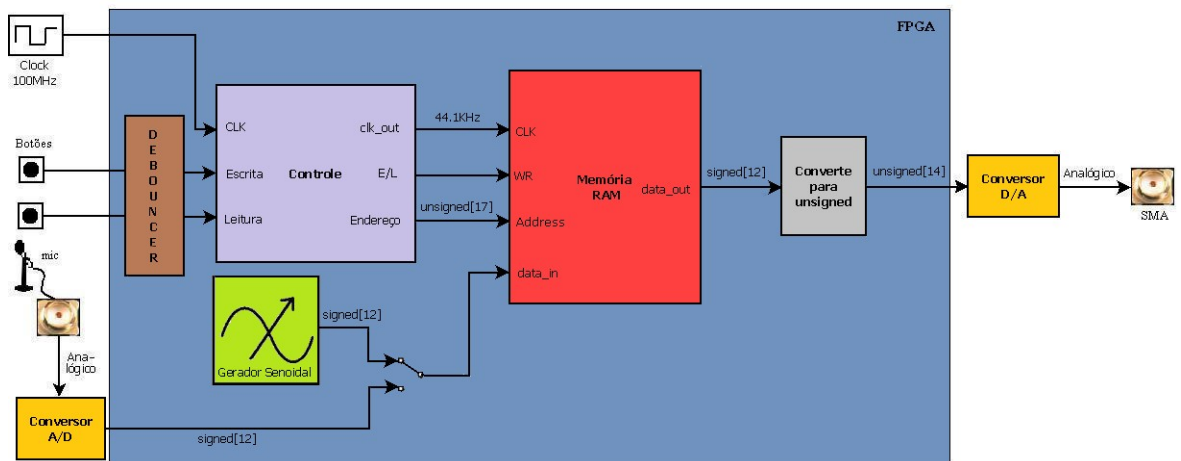


Figura 4 - Diagrama do circuito de armazenamento de sinal.

A memória RAM usada neste cenário possui palavras de 12 bits com 17 bits de endereçamento, e dispõe de uma porta para leitura e outra pra escrita de dados. O tamanho da palavra armazenada na RAM é determinada pela quantidade de bits das amostras na saída do conversor A/D de 12 bits. Os 17 bits de endereço são suficientes para armazenar até 3,9 segundos.

3.4 Gerador de sinal senoidal

Com o intuito de implementar um gerador com frequência e amplitude ajustável e boa qualidade espectral, foi criado circuito para controlar o funcionamento do IP NCO, utilizando os 4 botões disponíveis no kit. Estes botões são usados para controlar tanto a frequência como a amplitude de saída do gerador. A arquitetura de NCO utilizada foi usando o algoritmo CORDIC. Os parâmetros do NCO foram configurados de tal modo que o sinal de saída pudesse ser conectado a entrada do conversor D/A, mantendo uma boa precisão no valor em Hertz da frequência do gerador. Na implementação feita o controle da frequência é feito através do valor ϕ_{inc} , conforme mostra a Tab. 1.

Com o objetivo de simplificar o uso do gerador senoidal foram utilizados 4 botões (chaves *push-button*) e dois mostradores (*displays*) de sete segmentos para controlar o ganho e frequência, conforme mostra a Fig 5.

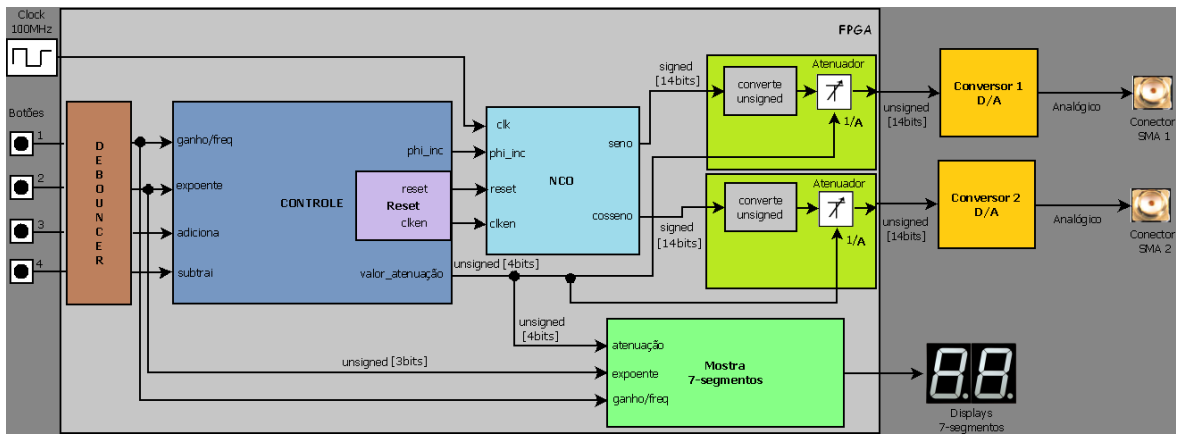


Figura 5 - Diagrama do Gerador de sinais senoidal.

O gerador criado mostrou-se funcional e com uma boa precisão de frequência e estabilidade. O controle de ganho e frequência permite ajustar os valores da frequência do sinal gerado na faixa de 30 kHz à 6,25 MHz com passos de 1 Hz, e o ganho entre 0 e 1 com passos de 0,1.

Tabela 1 - Relação entre o incremento do valor da frequência de saída do gerador e incremento do ϕ_{inc}

Δf_0	$\Delta \phi_{inc}$
1 Hz	43
10 Hz	429
100 Hz	4295
1 kHz	42950
10 kHz	429497
100 kHz	4294967
1 MHz	42949673

3.5 Filtros

O cenário realizado para teste dos filtros é mostrado na Fig. 6. Na entrada do filtro sob teste é possível conectar a soma de duas senoides com frequências diferentes, ou senoides isoladas. Tanto o sinal de entrada como a saída do filtro estão conectados aos conversores D/A para possibilitar a sua medição externa para verificar o funcionamento do filtro. Neste cenário foram utilizados dois botões para selecionar o envio de cada senoide para o bloco somador.

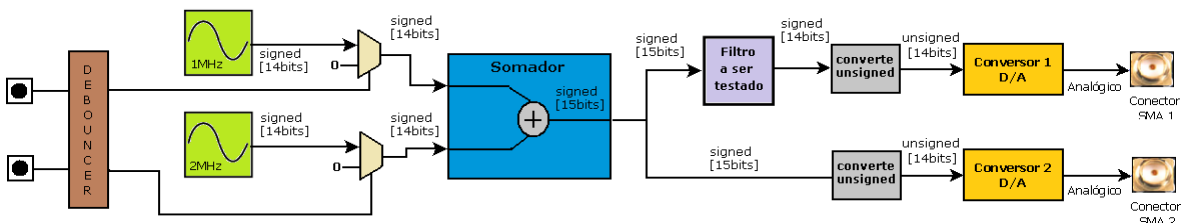


Figura 6 - Diagrama do circuito para teste dos Filtros.

Os dois geradores senoidais são implementados utilizando o NCO Altera com arquitetura ROM pequena, de forma a permitir gerar duas frequências diferentes. O bloco somador recebe os dois sinais de 14 bits vindos dos geradores e os soma resultando um sinal de 15 bits. O Circuito de anti-repique foi utilizado para evitar o repique das chaves.

O blocos conversores para *unsigned* transformam aos valores das amostras do tipo inteiro em complemento dois (*signed*) para inteiro positivo (*unsigned*).

3.6 Misturador de frequência

O circuito misturador criado recebe um sinal modulado AM através de um conversor A/D. Este sinal modulado é uma senoide de 20 kHz sobre uma portadora de 2 MHz obtida com o gerador de função MFG-4201A. Após ser convertido para digital, o sinal modulado é multiplicado por uma senoide criada por um gerador ajustável. Como a frequência inicial criada pelo gerador é de 1 MHz, o produto da multiplicação resulta em um sinal com portadoras em 1 MHz e em 3 MHz. Alterando a frequência criada pelo gerador, é possível notar o deslocamento do sinal produto.

O sinal resultante pode passar por filtros para eliminar frequências indesejadas. Neste cenário, o sinal é filtrado por um filtro passa-baixas e um filtro passa-faixa, de forma a demonstrar a eliminação dos componentes na faixa de 3 MHz. A Figura 7 mostra o diagrama do circuito criado.

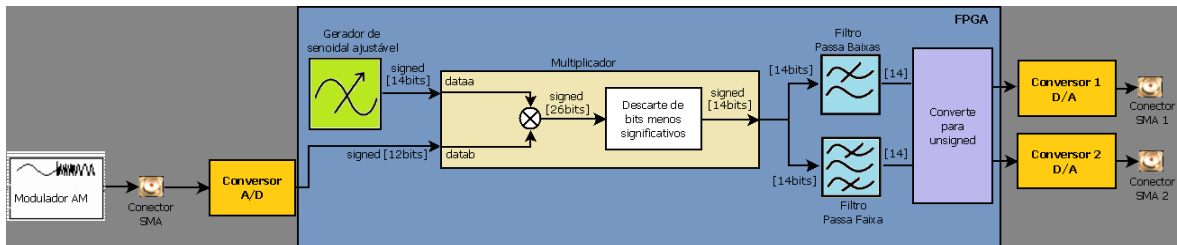


Figura 7 - Diagrama do circuito misturador de frequência.

3.7 Taxa de utilização do hardware no FPGA

Cada cenário foi implementado em FPGA usando o kit *DSP EP2S60* da ALTERA, e o resumo do hardware utilizado é apresentado nas tabelas 2 a 6. Nota-se uma baixa taxa de ocupação nas aplicações propostas, o que permitiria certamente o uso de chips de menor custo.

Tabela 2 - Hardware utilizado no cenário de aquisição e reprodução

Componentes	Utilizados	Disponíveis	% utilizada
ALUTs	827	48352	2%
Registradores lógicos dedicados	1031	48352	2%
Blocos de memória	335872	2544192	13%
Elementos de DSP	2	288	1%

Tabela 3 - Hardware utilizado no cenário de armazenamento de sinal

Componentes	Utilizados	Disponíveis	% utilizada
ALUTs	537	48352	1%
Registradores lógicos dedicados	439	48352	1%
Blocos de memória	1582080	2544192	62%
Elementos de DSP	4	288	1%

Tabela 4 - Hardware utilizado para criar o gerador senoidal

Componentes	Utilizados	Disponíveis	% utilizada
ALUTs	2295	48352	5%
Registradores lógicos dedicados	1727	48352	4%
Blocos de memória	0	2544192	0%
Elementos de DSP	4	288	1%

Tabela 5 - Hardware utilizado para testar um filtro FIR rejeita faixa com ordem 200

Componentes	Utilizados	Disponíveis	% utilizada
ALUTs	9178	48352	19%
Registradores lógicos dedicados	10288	48352	21%
Blocos de memória	426191	2544192	17%
Elementos de DSP	0	288	0%

Tabela 6 - Hardware utilizado para implementar o misturador

Componentes	Utilizados	Disponíveis	% utilizada
ALUTs	25406	48352	53%
Registradores lógicos dedicados	26550	48352	55%
Blocos de memória	213474	2544192	8%
Elementos de DSP	2	288	1%

4 Conclusões e trabalhos futuros

Este trabalho disponibiliza uma ferramenta didática que servirá como suporte a trabalhos futuros possibilitando implementar em tempo real, sistemas desenvolvidos no ambiente de desenvolvimento Matlab/Simulink envolvendo o processamento de sinais. A plataforma PS-FPGA possui a flexibilidade para permitir o desenvolvimento de novos módulos para a criação de sistemas mais complexos, tais como aplicações para imagem ou vídeo. A taxa de ocupação do chip FPGA, mostra que circuitos de complexidade superior podem ser implementados, de modo simples. Em relação ao kit *DSP EP2S60* da ALTERA utilizado, percebeu-se que o transformador de RF existente junto aos conectores externos SMA impede o uso dos conversores D/A e A/D para sinais de áudio. Em função disso sugere-se que novos estudos e módulos sejam implementados para explorar o CODEC de áudio existente no mesmo kit visando aplicações na faixa entre 20 Hz e 20 kHz. Diante dos resultados obtidos, sugere-se que se implemente circuito adicionais que transformem o módulo gerador senoidal em um equipamento de bancada para laboratório de eletrônica e telecomunicações, com boa qualidade e baixo custo. A implementação de equipamentos de análise espectral, usando o misturar de frequências e os filtros seletivos é também uma proposta que se mostra viável com o kit utilizado.

Agradecimentos

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro sem qual não seria possível desenvolver o presente trabalho.

Referências

ALTERA. **FIR Compiler - User Guide**. [S.l.], 2011. Disponível em http://www.altera.com/literature/ug/fircompiler_ug.pdf acesso em: julho, 2011.

ALTERA. **NCO MegaCore Function - User Guide**. [S.l.], 2011. Disponível em http://www.altera.com/literature/ug/ug_nco.pdf?GSA_pos=1&WT.oss_r=1&WT.oss=NCO%20user%20guide. acesso em: julho, 2011.

ESPINDOLA B. M.; PERES G. B.; MOECKE, M. Resumo: **Plataforma em FPGA para Processamento de Voz com Wavelet**. [S.l.], 2010. Disponível em <http://jornadacientificasul.blogspot.com/2010/12/anais-da-iii-jornada-ja-estao.html>. acesso em: julho, 2011.

LATHI, B. P. *Modern Digital and Analog Communication*. Terceira edição. [S.l.]: Oxford University Press, 1998.

PEDRONI, V. A. **Digital Electronics and Design with VHDL**. [S.l.]: The MIT Press, 2007.

SHENOI, B. A. **Introduction to Digital Signal Processing and Filter Design**. [S.l.]: Wiley-Interscience, 2006.

Responsabilidade de autoria

As informações contidas neste artigo são de inteira responsabilidade de seus autores. As opiniões nele emitidas não representam, necessariamente, pontos de vista da Instituição e/ou do Conselho Editorial do IF-SC.