

IMPLEMENTAÇÃO EM FPGA DE UMA CLASSE DE CÓDIGOS PRODUTO COM DECODIFICAÇÃO TURBO

Ivan Simões Gaspar¹, Dayan Adionel Guimarães²

Abstract — This work describes a project sponsored by LINEAR Equipamentos Eletrônicos S/A and carried out at INATEL. The project aimed to implement a turbo forward error correction scheme with a low complexity block turbo codec assembled in a popular low cost FPGA. The turbo decoding is based on a combination of Pyndiah's and Wagner's algorithms. Data rates of up to 80 Mbps were achieved. As a complementary result, the use of well-known and didactic simulation tools is explored, allowing the understanding of the specific class of product code and the proper FPGA implementation process. The codec was created with great flexibility using structural and behavioral models, easily translated later into the VHDL language.

Index Terms — Block Turbo Codes, FPGA.

I-INTRODUÇÃO

Sistemas de múltiplo acesso com multiportadoras são potenciais candidatos às futuras gerações de comunicação sem fio, oferecendo robustez contra desvanecimentos seletivos com a possível combinação de diversidade temporal e em frequência, contornando o uso de complexa equalização no controle de ISI (*Intersymbol Interference*), além de flexibilidade na composição do espectro final (similar ao espectro de Nyquist). Técnicas de apoio vêm sendo desenvolvidas para firmar esta tendência. Entre elas, estratégias de codificação de canal eficientes e simples representam um desafio.

Este artigo descreve o desenvolvimento em hardware de um *codec* turbo destinado ao aprimoramento do sistema MC-DS-CDMA (*Multi-Carrier Direct Sequence Code Division Multiple Access*) sugerido por [2], substituindo-se o código de repetição nele existente por um código produto multidimensional decodificado de forma iterativa [1]. Usou-se como dispositivo de implementação um FPGA (*Field Programmable Gate Array*) da família Cyclone (EP1C6T144C8), fabricado pela ALTERA. Esta tecnologia de custo relativamente baixo permitiu que taxas úteis até 10 Mbps pudessem ser atingidas em uma placa protótipo.

O esquema de codificação proposto é constituído de uma classe de código produto multidimensional obtida com a concatenação serial generalizada de um mesmo código componente, um exemplo típico de GCC (*Generalized Code Concatenation*) [3]. A característica chave deste arranjo é que é possível construir para ele uma estrutura de

decodificação de entrada suave aplicando o algoritmo de decodificação por distância mínima de Wagner [4].

A complexidade de decodificação resultante em cada dimensão é semelhante à de um código produto de paridade simples [5][6] e o desempenho é equiparável àquele obtido por meio de procura exaustiva. Procurando se reduzir ainda mais a complexidade, uma forma adaptada do algoritmo de decodificação iterativa de Pyndiah [7] é aplicado ao arranjo. Neste último caso algumas considerações para a quantização dos coeficientes de ponderação são investigadas utilizando-se como ferramentas de simulação os softwares VisSim/Comm e Mathcad.

II-ELABORAÇÃO DO ESQUEMA DE CODIFICAÇÃO

O código componente possui taxa $k/n=1/2$ e é formado por um arranjo de partição de conjunto obtido através da combinação de um código de repetição com um código de paridade simples. Seja c_1 uma palavra-código do código de repetição $C_1 = (n/2, 1, n/2)$ e c_2 uma palavra-código do código de paridade simples $C_2 = (n/2, n/2-1, 2)$. A palavra-código c do código não-sistemático $C = (n, k, d_{\min}) = (n, n/2, 4)$ é determinada por

$$c = [01]c_1 \oplus [11]c_2 \quad (1)$$

onde a soma \oplus é em GF(2) e o produto $[01]c_1$ é calculado substituindo-se um 0 em c_1 por 00 e um 1 por 01. O mesmo é feito para $[11]c_2$, onde agora um 1 torna-se 11. O bit mais à esquerda da palavra de mensagem é utilizado como entrada para o código de repetição e os demais são utilizados como informação para o código de paridade simples.

A formação do código tal como descrita sugere uma estrutura serial e essa possibilidade foi investigada com o auxílio do software de simulação VisSim/Comm. Um dos principais benefícios da utilização deste programa no projeto se refere à simplicidade de criação de blocos hierárquicos parametrizáveis, à transparência da depuração e verificação do funcionamento e à similaridade com que tais blocos são posteriormente convertidos em VHDL (*Very high speed integrated circuit Hardware Description Language*).

Cada parte do diagrama de blocos é escrita como um processo concorrente, e mesmo detalhes de alinhamento de sinais de controle podem ser reproduzidos no VisSim/Comm (blocos de “*delay2*” na Figura 1).

¹ Linear Equipamentos Eletrônicos S/A, Praça Linear, 100, Depto. de Projetos, 37.540-000, Santa Rita do Sapucaí, MG, Brasil, igaspar@linear.com.br
² Inatel – Instituto Nacional de Telecomunicações, Av. João de Camargo, 510, 37.540-000, Santa Rita do Sapucaí, MG, Brasil, dayan@inatel.br

A partir do recebimento do primeiro bit de mensagem amostrado por um *clock* equivalente a duas vezes a taxa de entrada (um requisito como taxa de saída deste código componente), cada um dos demais bits serão repassados a saída e formatados após meio intervalo de bit de entrada. Durante o intervalo do primeiro bit de uma nova palavra de informação, encerra-se o processamento da paridade da anterior e se inicia o cálculo da paridade para a nova palavra. Se o primeiro bit de entrada for 0, a partição do conjunto corresponde aos *dibits* 00/11, nenhuma atitude de formatação adicional no código de paridade simples é necessária. No outro caso, primeiro bit igual a 1, após metade do tempo de bit de entrada o pulso de saída é invertido, partição 01/10. Essa construção parametrizável foi posteriormente traduzida em VHDL utilizando basicamente os mesmos processos de lógica mostrados na Figura 1 (um contador de *n* bits codificados, um detector de paridade, etc.). Embora os casos com melhor desempenho sejam conhecidos apenas três (8,4), (12,6,4) e (16,8,4) [1], a construção genérica dos blocos viabiliza a sua inclusão em bibliotecas de programação.

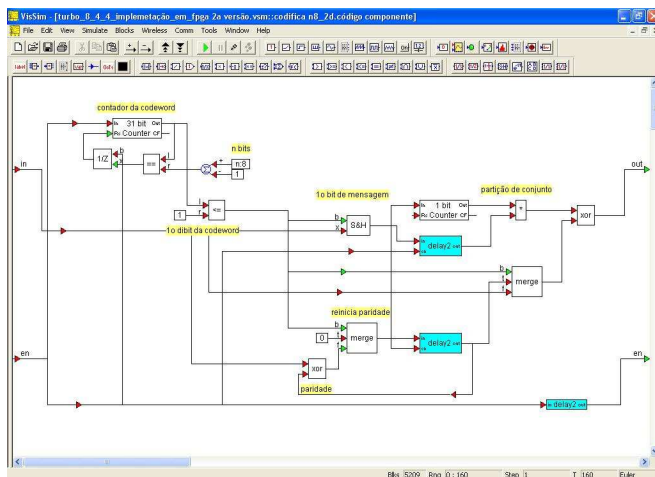


FIGURA. 1

ESTRUTURA DO CÓDIGO COMPONENTE IMPLEMENTADA NO VISSIM/COMM

Utilizando o mesmo código não-sistemático *C* como código componente em cada uma das *D* dimensões, obtém-se um código produto de comprimento n^D , taxa $(1/2)^D$ e distância mínima 4^D .

O codificador bidimensional 2D mostrado na Figura 2 usa $D = 2$ e $k = 4$ e é denotado por $(8,4)^2$. Na saída do primeiro estágio de codificação cada grupo de 4×4 bits de mensagem resulta em um bloco bidimensional 4×8 . A próxima etapa de codificação é obtida por meio do uso de um entrelaçador (*interleaver*) construído com parte da memória interna do FPGA, operando nesses 4×8 bits. Essa memória é dividida em duas partes: enquanto uma é usada para armazenar a saída do primeiro codificador linha-a-linha, a segunda entrega coluna-a-coluna os dados para o próximo codificador.

O controle de endereços de escrita segue uma progressão linear de 0 a $2 \times N_c \times N_l - 1$ (N_c é o número de colunas e N_l o número de linhas do bloco entrelaçador). Enquanto os dados são escritos na primeira parte da memória (faixa de 0 à $N_c \times N_l - 1$), cada endereço de leitura, E_L , é determinado conforme a regra

$$E_L = C_c + (C_l \times N_c) + (N_c \times N_l) \quad (2)$$

onde C_c e C_l são os contadores de coluna e de linha, respectivamente, mostrados na Figura 3.

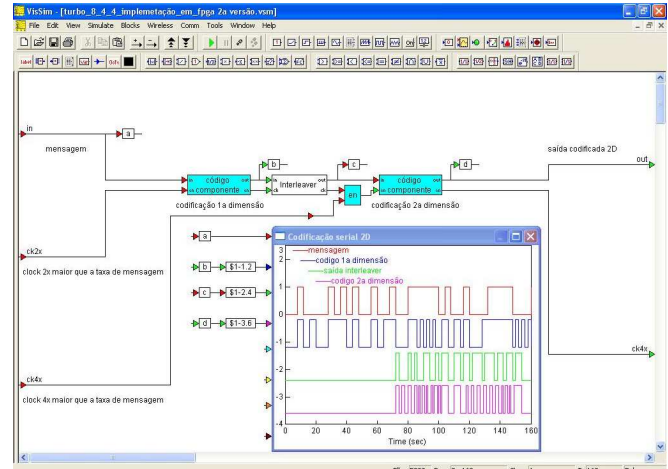


FIGURA. 2

ESTRUTURA SERIAL DE CODIFICAÇÃO EM 2D. O BLOCO “CÓDIGO COMPONENTE” ESTÁ MOSTRADO EM DETALHES NA FIGURA 1

Quando a faixa de escrita vai de $N_c \times N_l$ a $2 \times N_c \times N_l - 1$, ignora-se o termo $N_c \times N_l$ na equação (2). A saída resultante é um arranjo bidimensional 16×16 .

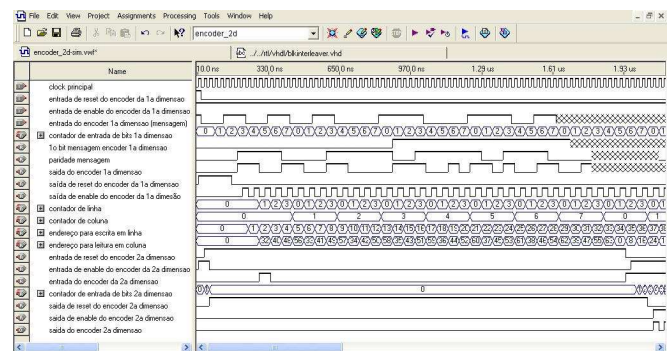


FIGURA. 3

SINAIS DE ENTRADA E SAÍDA DOS COMPONENTES DO CODIFICADOR 2D SIMULADOS COM A FERRAMENTA DE SÍNTESE QUARTUS II DA ALTERA

Cada componente do circuito de codificação foi provido de entradas e saídas de controles de habilitação e de reinício, dispensando a necessidade de se criar um controle central para integração dos blocos. Essa medida mantém a

modularidade do conjunto e permite que a construção para um D qualquer seja uma simples concatenação dos sinais de entrada e saída dos blocos de códigos componentes e entrelaçadores devidamente parametrizados.

A frequência de operação do codificador 2D implementado de forma serial é de até 200 MHz, permitindo taxas úteis de entrada até 50 Mbps. A montagem do projeto ocupou um recurso de 77 elementos lógicos e 64 bits de memória (menos de 1% da lógica disponível).

Uma implementação alternativa por meio do mapeamento combinacional para palavras código pequenas é também possível: mensagens de k bits endereçando uma LUT (*Look-Up Table*) com n bits de saída podem operar com *clocks* acima de 200 MHz. Esta forma de construção paralela ocupa maior quantidade de lógica, mas seria particularmente conveniente se aplicada no sistema MC-DS-CDMA de [2] quando o número de bits, M , na saída do conversor série/paralelo fosse adequadamente escolhido como k^D , e o parâmetro $M \times S$ como $(n, k)^D$.

III-DECODIFICAÇÃO DO CÓDIGO COMPONENTE

O Algoritmo de Wagner, inicialmente chamado de Código de Wagner [4], foi proposto para decodificação de códigos de verificação de paridade. Contudo, sua utilização pode ser estendida para quaisquer casos em que a decodificação possa ser baseada em processos de verificação de paridade, como é o caso de alguns códigos de Reed Muller e de Hamming.

Conforme [1], as probabilidades *a-posteriori* $p(x_1/y)$ e $p(x_2/y)$ de cada dígito y recebido são calculadas e a estimativa dos dígitos transmitidos é feita de acordo com o critério de máxima verossimilhança ML (*Maximum Likelihood*), ou seja, é escolhido x_1 se $p(x_1/y) > p(x_2/y)$ e x_2 caso contrário. Se $p(x_1/y) = p(x_2/y)$ escolhe-se x_1 ou x_2 com igual probabilidade, arbitrariamente. O critério ML é implementado a partir da distância euclidiana quadrática entre a amplitude do sinal recebido na saída do correlator e a amplitude do sinal transmitido normalizada em relação à média do sinal recebido, ou seja, escolhe-se x_1 se $(y - x_1)^2 < (y - x_2)^2$ e x_2 caso contrário. Tendo-se decidido pelos n dígitos, verifica-se a paridade. Quando correta, a palavra estimada é considerada como a mais provável. Em caso contrário é invertido o dígito mais “duvidoso” da palavra anteriormente estimada, formando a estimativa final. O dígito mais duvidoso é aquele que apresenta a menor diferença $|(y - x_1)^2 - (y - x_2)^2|$ onde $|x|$ é o módulo de x .

A primeira versão do decodificador implementado em FPGA foi concebida para operar serialmente, de forma que durante o carregamento de uma nova palavra se processasse a saída decodificada da palavra anterior. Basicamente duas estruturas paralelas são alimentadas com os dados suaves quantizados em 5 bits. De forma sincronizada, dois a dois os bits são agrupados para que as métricas (associadas às distâncias Euclidianas) das seções em cada ramo sejam calculadas. À medida que as decisões são tomadas, um acumulador de métricas em cada é atualizado. O

decodificador aplica a regra de Wagner [4] para o código de paridade simples de comprimento $n/2$, sobre o alfabeto binário {00, 11} e sobre o alfabeto {01, 10}, obtendo as decisões \hat{c} e \hat{c}' . Comparando as métricas entre a palavra-código recebida, r , e as decisões \hat{c} e \hat{c}' escolhe-se como decisão final a palavra que estiver mais próxima de r .

Experimentou-se no projeto um caso particular para definição do cálculo das métricas: considerou-se na entrada do circuito ADC (*Analog to Digital Converter*) um sistema de AGC (*Automatic Gain Control*) que adapta a faixa dinâmica do sinal a ser quantizado, limitando-a a um valor máximo. Os valores limites desta faixa, para uma quantização com 5 bits, são os inteiros $+2^4-1$ e -2^4 . Estes são os valores utilizados como referência para os cálculos das métricas, ou seja, calculam-se as distâncias em relação a estes máximos e não em relação a valores médios³.

Posteriormente elaborou-se uma construção paralela de decodificação totalmente combinacional (Figura 4). Os bits quantizados são aplicados a um arranjo de LUT's (*Look-Up Tables*) contendo comparadores e multiplexadores. Dois a dois os sinais são comparados às métricas máximas dos díbits 00/11 e 01/10. A decisão final é disponibilizada em um único ciclo de *clock*.

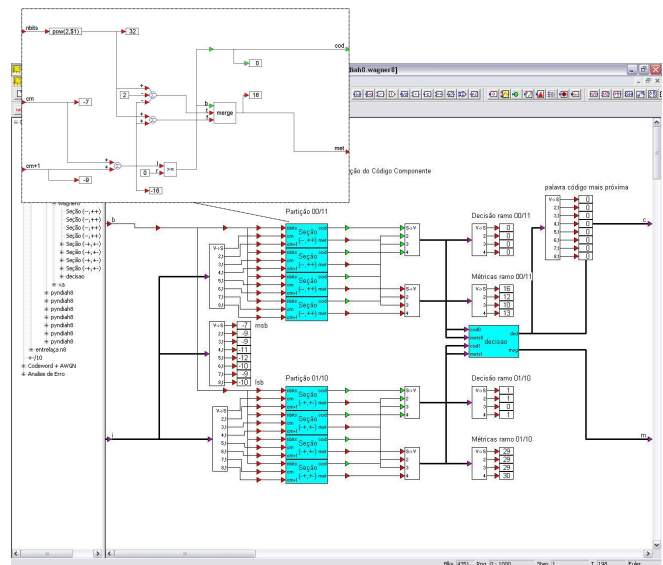


FIGURA. 4

CONSTRUÇÃO PARALELIZADA DO DECODIFICADOR DO CÓDIGO (8,4)

Em ambas as formas de construção testadas (serial e paralela) o desempenho obtido é o mesmo. A diferença básica está na quantidade de lógica, na velocidade final de operação e na conveniência quanto ao seu emprego em um processo iterativo. Ocupando 157 elementos lógicos, o arranjo serial pode operar com taxas de até 63 Mbps (126 MHz de *clock*), enquanto o decodificador paralelo, com 257

³ Esses valores médios se referem aos níveis de referência de sinal correspondentes aos pontos da constelação (x_1 e x_2).

elementos lógicos, pode operar com *clock* de até 33 MHz. Embora com maior tempo de propagação, o circuito combinacional tem aproximadamente o dobro do desempenho da versão serial, pois todos os bits decodificados são disponibilizados a cada ciclo de *clock*. Neste caso pode-se decodificar taxas úteis de até $4 \times 33 \times 10^6 = 132$ Mbps. Os resultados práticos validam a consideração adotada, coincidindo com os valores obtidos com simulação.

IV-DECODIFICAÇÃO ITERATIVA

R. M. Pyndiah apresenta em [6] um algoritmo sub-ótimo para decodificação iterativa de códigos de bloco baseados em códigos produto. Esse algoritmo utiliza um decodificador SIHO (*Soft-Input Hard-Output*) baseado no Algoritmo 2 de Chase [8], seguido de cálculos para obter informações de confiabilidade (saída suave) a partir das decisões abruptas do decodificador. A proposta apresenta uma boa solução de compromisso entre desempenho e complexidade, sendo bastante atrativa para aplicações práticas.

Em [1] utiliza-se a idéia básica do algoritmo de Pyndiah, porém substituindo o algoritmo de Chase do decodificador SIHO pelo algoritmo de Wagner. Restringe-se com isso o rol de códigos que podem se valer dessa nova combinação, em relação à utilização do algoritmo de Chase que se aplica a qualquer código de bloco linear, porém tendo como benefício uma esperada redução da complexidade de implementação do decodificador, em comparação à implementação com o algoritmo de Chase.

No processo de decodificação iterativa implementado, os valores *a-priori* para cada símbolo são considerados nulos antes da primeira iteração. A saída do decodificador SISO é composta pelos valores *a-posteriori* de todos os bits codificados, mais a informação extrínseca obtida pelo processo de decodificação. Esse valor de informação extrínseca é realimentado à entrada do decodificador SISO como o valor da log-verossimilhança *a-priori* para a próxima iteração.

O diagrama de blocos da Figura 5 representa a operação de decodificação no *j*-ésimo passo, onde o máximo valor de *j*, j_{max} , é o resultado da multiplicação do número de iterações, *I*, pela dimensão do código, *D*. O arranjo **R** contém todos os n^D símbolos recebidos. A expressão “decodificação em uma das dimensões” que aparece na figura significa a decodificação de n^{D-2} arranjos bidimensionais $n \times n$ na direção de uma das dimensões. A decodificação de um arranjo desse tipo consiste da decodificação de *n* linhas (ou *n* colunas) e, portanto, a decodificação em uma das dimensões implica na utilização do algoritmo SISO n^{D-1} vezes.

Os fatores de escala $\alpha(j)$ e $\beta(j)$ fornecem um meio de ponderação no processo iterativo. A variação desses fatores ocorre de forma exponencial e linear, respectivamente. Como a decisão parcial sobre dos bits em cada dimensão é revogável, à medida que sua confiabilidade é experimentada em cada iteração podem ocorrer sucessivas alternâncias de

polaridade. A curva da Figura 6 ilustra os resultados de simulação apresentados em [1] para a configuração de codec adotada para implementação.

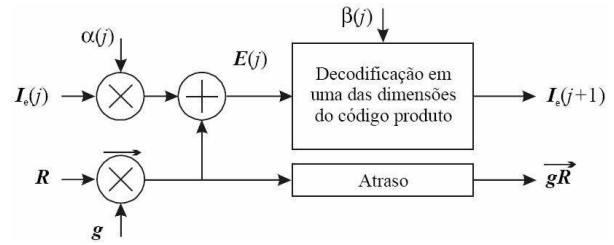


FIGURA. 5

ESQUEMA INICIALMENTE PROPOSTO PARA DECODIFICAÇÃO ITERATIVA

Os fatores de ponderação foram quantizados de forma que a cada iteração a entrada suave para decodificação do código componente possuísse 5 bits. A implementação final do decodificador tubo ocupou praticamente toda a área disponível do dispositivo (ver Figura 7), podendo efetuar cada iteração parcial do bloco codificado a uma frequência de 40 MHz. Considerando que 2 iterações parciais correspondem a uma iteração completa e que 4 iterações sejam efetuadas, a cada 8 pulsos de *clock* podem-se ter disponíveis os 16 bits de mensagem decodificados na palavra código de 64 bits, uma taxa útil de $16 \times 40 \times 10^6 / 8 = 80$ Mbps. A implementação prática testada proporcionou desempenho distante cerca de 1 dB da curva simulada, degradação que se acredita ter como um dos fatores a quantização de $\alpha(j)$ e $\beta(j)$.

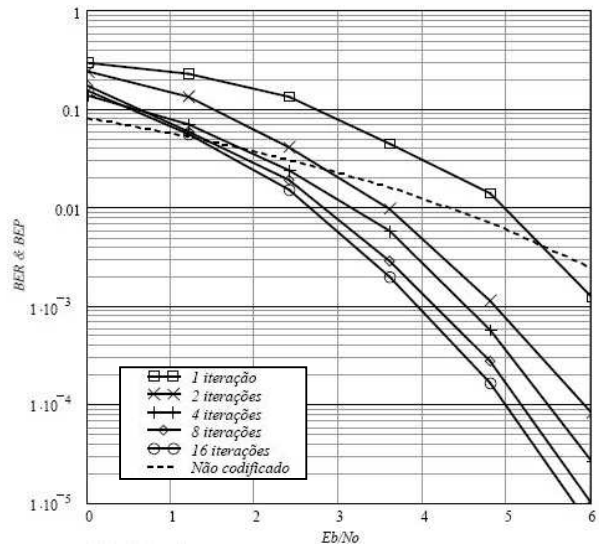


FIGURA. 6

EVOLUÇÃO NO DESEMPENHO DO CÓDIGO PRODUTO 2D COM COMPONENTES (8,4,4) EM CANAL AWGN, EM FUNÇÃO DO NÚMERO DE ITERAÇÕES.

A Figura 8 ilustra o teste em bancada realizado com o código turbo implementado. Um gerador HAMEG foi

utilizado como fonte alternativa de *clock* para o sistema. Uma seqüência de bits aleatórios, gerada para teste de BER no equipamento SMIQ 04B de fabricação ROHDE & SCHWARZ, é injetado no circuito. O sinal é codificado e contaminado por um ruído Gaussiano sintetizado vetorialmente no FPGA. O resultado dessa interferência foi exteriorizado para demonstração com o auxílio de um conversor D/A presente na placa. Os efeitos de degradação podem ser visualizados com o auxílio do recurso de histograma do osciloscópio digital Agilent 54832B Infiniium.

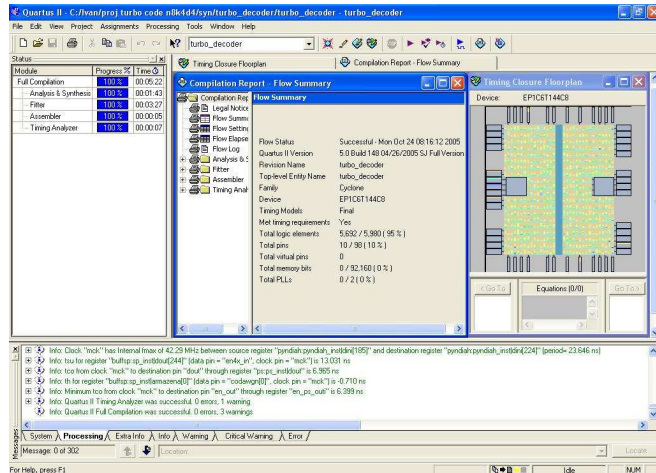


FIGURA. 7

SISTEMA DE DECODIFICAÇÃO ITERATIVA IMPLEMENTADO EM FPGA

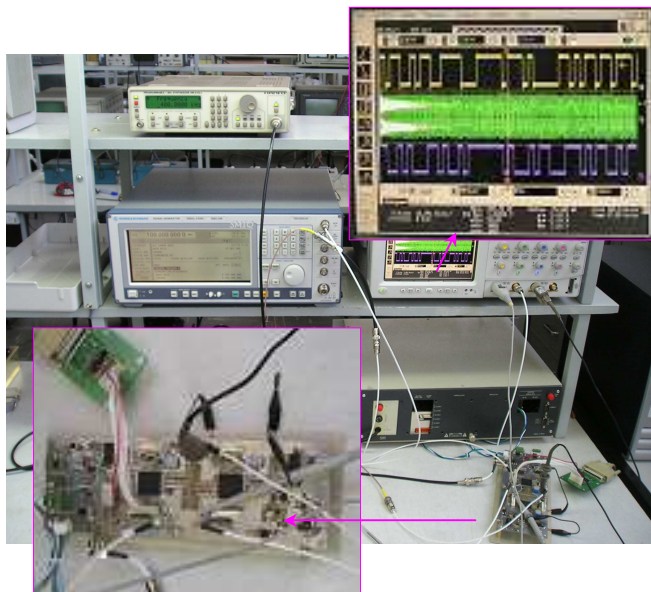


FIGURA. 8

TESTE EM BANCADA DA DECODIFICAÇÃO TURBO PARA O CÓDIGO $(8,4)^2$

V-CONCLUSÃO

Este artigo apresentou o processo de desenvolvimento e implementação em FPGA de uma classe de códigos produto de dimensão D , comprimento n^D , taxa $(1/2)^D$ e distância mínima 4^D , proposta inicialmente em [1] para sistemas CDMA Multiportadora, para $D = 2$ e $n = 8$.

Os resultados obtidos com a implementação do *codec* turbo distaram 1dB do desempenho previsto em simulação, perda atribuída às considerações de truncamento adotadas na implementação.

Taxas de até 80 Mbps foram testadas nesse esquema de codificação, um resultado bastante atrativo dado o baixo custo e limitado desempenho do FPGA utilizado (Altera EP1C6T144C8). Vale ressaltar que o uso de FPGAs de custo ainda reduzido, mas com melhor desempenho, pode proporcionar taxas úteis ainda maiores.

Ressalta-se ainda que a latência do decodificador foi limitada mais pelos tempos necessários à “serialização” e “de-serialização” dos dados codificados do que pelo processo de decodificação iterativo. Re-estudos em termos da implementação podem melhorar ainda mais o desempenho do *codec* implementado.

AGRADECIMENTO

Os autores agradecem a colaboração dada pela empresa Linear, na pessoa de seu Presidente, Sr. José de Souza Lima, apoiando este projeto por meio da disponibilização de recursos de laboratório e de tempo.

REFERÊNCIAS

- [1] Guimarães, D., A., "Uma classe de códigos produto e sua decodificação turbo aplicada em um sistema CDMA multiportadora", *Tese de Doutorado*, Unicamp, Campinas, SP, junho 2003.
- [2] Sourour, E., e Nakagawa, M., "Performance of Orthogonal Multicarrier CDMA in a Multipath Fading Channel", *IEEE Transactions on Communications*, Vol. 44, No. 3, março de 1996, pp. 356-367.
- [3] Bossert, M. "Channel-Coding for Telecommunications". *John Wiley & Sons, Ltd.*, England, July/2000.
- [4] Silverman, R., A. e Balser, M., "Coding for Constant-Data-Rate Systems", *IRE Transactions on Information Theory*, PGIT-4, 1954, pp. 50-63.
- [5] Sklar, B., "A Primer on Turbo Code Concepts". *IEEE Communication Magazine*, dezembro de 1997, pp.94-101.
- [6] Rankin, D., M. e Gulliver, T., A., "Single Parity Check Product Codes", *IEEE Transactions on Communications*, Vol. 49, No. 8, agosto de 1998, pp. 1354-1362.
- [7] Pyndiah, R., M., "Near-Optimum Decoding of Product Codes: Block Turbo Codes", *IEEE Transactions on Communications*, Vol. 46, No. 8, agosto de 1998, pp. 1003-1010.
- [8] Chase, D., "A Class of Algorithms for Decoding Block Codes With Channel Measurement Information". *IEEE Transactions on Information Theory*, Vol. IT- 18, No. 1., Janeiro de 1972, pp. 170-182