

PROJETO DE UM CONVERSOR ANALOGICO A DIGITAL ASSINCRONO PARA SINAIS QUASE ESTACIONARIAS UTILIZANDO FPGA

RAUL REQUENA TAYPE

Facultad de Ingenieria Electrica y Electronica, Universidad Nacional de Ingenieria, Lima-Perú
E-mails: recyprus@hotmail.com

JUAN M. MAURICIO VILLANUEVA, RAIMUNDO CARLOS S. FREIRE

Departamento de Engenharia Eletrica, Universidade Federal de Campina Grande, Campina Grande-Brasil
E-mails: jmauricio@ee.ufcg.edu.br, freire@dee.ufcg.edu.br

Abstract— The conversion of an analog into a digital signal is of great importance for several applications of instrumentation systems and measurement which is performed generally of periodic manner in time. Nevertheless, for applications such as environmental monitoring, weather, environmental, precision farming, etc., the characteristics of signals are quasi stationary, i.e. with small variations in time; it is proposed the use of analog to digital converters using the asynchronous sampling. For this purpose, this paper presents the design of an analog-digital converter using asynchronous FPGA ALTERA Stratix III. Finally, simulation results are presented for 4-bit resolutions reaching a frequency of 5 MHz with a power consumption of 438 μ W.

Keywords— Asynchronous ADC, FPGA, Quasi Stationary Signals

Resumo— O processo de conversão de sinais analógicos para digitais é uma etapa de grande importância para diversas aplicações de sistemas de instrumentação e medição, a qual é realizada geralmente a intervalos iguais no tempo. Entretanto, para aplicações tais como a monitoração ambiental, climática, ecológica e em agricultura de precisão, cujas características dos sinais são do tipo quase estacionárias, isto é, com pequenas variações no tempo, pode-se usar os conversores analógico/digital com amostragem assíncrona. Para este propósito, neste trabalho apresenta-se o projeto de um conversor análogo-digital assíncrono utilizando o FPGA Stratix III de ALTERA. São apresentados, também, resultados de simulação para resoluções de 4 bits e frequência de 5 MHz, com um consumo de potência de 438 μ W.

Palavras-chave— Conversores ADC Assíncrono, FPGA, Sinais Quase Estacionarias.

1 Introdução

O processo de conversão de sinais analógicos para digitais (ADC) é uma das etapas mais importantes em sistemas de instrumentação, medição e processamento de sinais. Geralmente, os ADC apresentam parâmetros tais como frequência de amostragem (usualmente constante), SNR de quantização, resolução em bits, potência consumida, etc. É possível também definir períodos de amostragem não regulares, que são mais próximos à natureza do sinal, os quais, dependendo da janela temporal da análise, podem apresentar características quase estacionarias, isto é, um comportamento sem uma variação significativa da amplitude no tempo. Tais aplicações são apresentadas em sistemas de monitoração ambiental, climática, ecológica e em agricultura de precisão. Desta maneira, um ADC com amostragem assíncrona aperfeiçoa o processo de aquisição, processamento e transmissão do sinal (Akopyan, 2006).

Além disso, em muitas aplicações, os pontos de aquisição dos sinais são distribuídos em uma região bastante ampla. Esses casos são típicos de utilização de redes de sensores sem fio (WSN-*Wireless Sensor Networks*). Um dos compromissos no projeto de WSN consiste em definir considerações de eficiência

energética para os módulos de transmissão, recepção, aquisição e processamento de sinais. Desta maneira o gerenciamento eficiente da energia é realizado identificando processos internos não necessários e redundantes, como o processo de sobre amostragem de sinais quando é usada uma frequência de amostragem superior a duas vezes a frequência máxima do sinal (Chinh, 2010).

Diversas abordagens para o projeto do ADC assíncrono têm sido desenvolvidas, entre elas, foi utilizadas técnicas de modulação sigma-delta, com um comparador e um conversor DA, entretanto, esta solução não é puramente assíncrona devido a que se requer um relógio para a conversão DA. Outra abordagem utiliza um arranjo de circuitos digitais para a ativação dos comparadores em funcionamento, entretanto estes circuitos digitais tem um elevado consumo de energia. Também, foram desenvolvidos projetos baseados em FPGA (*Field-Programmable Gate Array*) mostrando que esta ferramenta é conveniente para o projeto de ADC assíncronos (Matić, 2010; Akopyan, 2006; Huot, 2005).

Neste trabalho, apresenta-se o projeto de um Conversor Analógico-Digital Assíncrono utilizando FPGA. Para este propósito, apresentam-se definições preliminares para a elaboração do projeto do ADC assíncrono, tais como, a amostragem não uniforme e modulação sigma delta. Mostra-se também o procedimento de implementação do ADC assíncrono

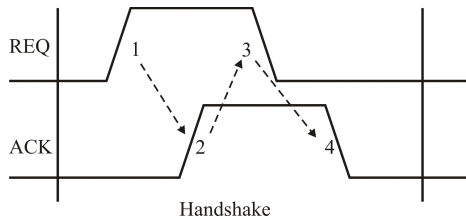


Figura 1. Sinalização a quatro fases.

em um FPGA STRATIX III de ALTERA. Finalmente, são apresentados resultados de simulação para um sinal quase estacionário, para diferentes valores de resolução em bits e consumo de potência.

2 Definições Preliminares

2.1 Projeto Digital Assíncrono

Na saída dos circuitos digitais síncronos e assíncronos os sinais são binários assumindo valores “uns” e “zeros”. Os circuitos síncronos usam pulsos de um relógio (*clock*) master de referência que não é usado nos casos dos circuitos assíncronos, nos quais se utilizam diferentes modelos de sincronização para a comunicação e sequência das operações (Sparso, 2001). Neste trabalho foi usado o protocolo de comunicação tipo *handshaking* que utiliza dois sinais de controle REQ e ACK, baseada na sinalização a quatro fases.

A sinalização a quatro fases é chamada também de sinalização com retorno a zero (RZ) e utiliza a borda positiva dos sinais de controle. Desta maneira, após cada comunicação, os sinais de controle tem que retornar ao nível baixo, como se ilustra na Figura 1. Esta sinalização apresenta melhores características para sua implementação, devido à disponibilidade de ferramentas padrões de projeto, as quais têm sido desenvolvidos para a manipulação de níveis lógicos de bordas positivas.

Como os sinais de controle não transportam dados, uma vez definidos os protocolos de comunicação se define o mecanismo para realizar a transferência dos dados entre o transmissor e o receptor baseado na forma *dual rail*, que utiliza duas linhas por bit de informação, uma para cada estado possível dos dados a transmitir (“0” ou “1”), como se ilustra na Figura 2. Assim, para o projeto do ADC assíncrono fez-se uso de modelos baseados nos valores de retardos das portas lógicas e das linhas de comunicação, para as linhas de controle e linhas de dados.

Como as linhas de controle (REQ e ACK) não

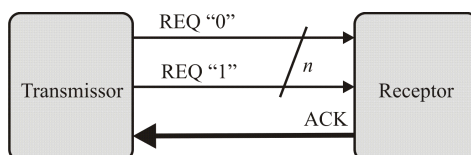


Figura 2. Transmissão DUAL RAIL.

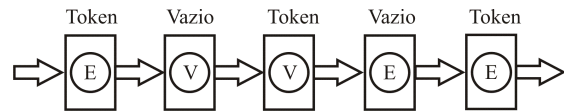


Figura 3. Sinalização a quatro fases.

permitem a transferência de dados entre módulos assíncronos, utilizou-se o conceito de TOKEN, que é representado por um círculo cheio (válido) ou vazio (não válido) dentro do elemento de memória onde está armazenado os dados. Os tokens podem passar de um latch ao seguinte e podem ser divididos, eliminados e reunidos com outros tokens em palavras lógicas.

Desta maneira, a regra geral para armazenar e transmitir tokens é: Um latch (i) pode aceitar um token (vazio “E” ou valido “V”) de seu predecessor ($i-1$) se e somente se seu sucessor ($i+1$) aceitou e armazenou o token do latch anterior (i) (Sparso, 2001). Na Figura 3 se ilustra a sequência de estados de transmissão dos tokens.

No projeto de circuitos digitais assíncronos, utilizam-se portas básicas orientadas à resposta de eventos em suas entradas, tais como XOR (ou exclusivo), ELEMENTO C (uma transição ocorre na saída somente após haver uma transição em cada uma de suas entradas), DELAY (usado para prover um retardo puro sobre as linhas de sinais de controle) e SELECT-K (recupera estados lógicos) (Sparso, 2001).

No âmbito do desenvolvimento de circuitos digitais assíncronos, um elemento importante é a micropipeline. Adotou-se neste trabalho a teoria assíncrona de sinalização por eventos (o princípio geral não distingue entre as bordas de subida ou descida). O mecanismo permite manipular sinais de controle REQ e ACK, elementos de armazenamento (*latch*) e blocos combinacionais, como se ilustra na Figura 4. A configuração e lógica de funcionamento requerido são realizadas usando uma cadeia de elementos “C”s e inversores para formar laços ao redor do fluxo de eventos, cuja dinâmica é: Se dois elementos tem o mesmo estado, este se mantém, mas se dois elementos têm estados diferentes, se copia o estado do elemento (i) no elemento ($i+1$) (Sutherland, 1989).

2.2 Amostragem não uniforme

Para o caso de sinais quase estacionarias é possível realizar a amostragem cada vez que o sinal cruza um nível de referência, como se ilustra na Figura 5. Desta maneira a informação sobre o sinal está contida nos intervalos de tempo entre cada amostra e nas direções de cada cruzamento. Com este procedimento se reduz a taxa de atividade, complexidade, envio de informação, espaço de armazenamento e consumo de potência.

Como condição para a reconstrução posterior do sinal pode-se usar o teorema de Shannon generalizado, definido por:

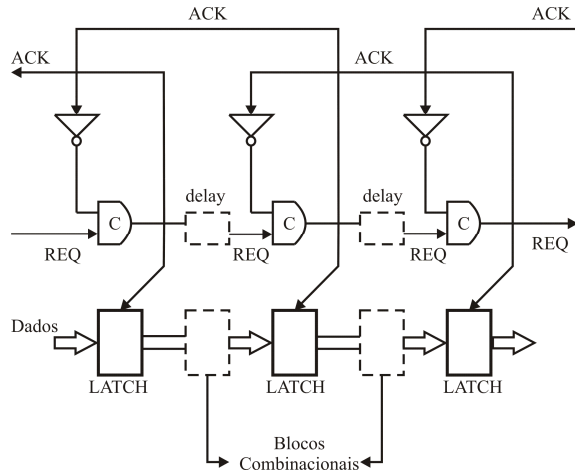


Figura 4. Arquitetura Micropipeline.

$$[f_{amostragem}]_{media} \geq 2f_{max} \quad (1)$$

Em que: $[f_{amostragem}]_{media}$ é a frequência média de amostras tomadas e f_{max} é a frequência máxima do sinal de entrada.

Para uma amostragem por cruzamento de nível de M bits se requer $2^M - 1$ níveis de referência de cruzamento. Considerando-se um sinal de entrada V e uma faixa dinâmica ΔV , o passo entre cada nível de referência é definido por:

$$q = \frac{\Delta V}{2^M} \quad (2)$$

Para não se perder informações, a taxa de variação da tensão do sinal de entrada com relação ao tempo deve ser menor ou igual à resolução q dividida pelo retardo de processamento δ .

$$\left| \frac{dV}{dt} \right| \leq \frac{q}{\delta} \quad (3)$$

Com o valor do retardo de processamento pode-se encontrar a frequência máxima do sinal de entrada. A partir da expressão (3), com o sinal V limitado em largura de banda de $[0, f_{max}]$, utilizando o teorema de Bernstein (Allier, 2005), encontra-se:

$$\left| \frac{dV}{dt} \right| \leq 2\pi \cdot f_{max} \cdot \Delta V \quad (4)$$

A partir de (2), (3) e (4) pode-se encontrar o retardo de processamento máximo, que é dado por:

$$\delta_{max} = \frac{1}{2\pi \cdot f_{max} \cdot 2^M} \quad (5)$$

A partir de (5) encontra-se a frequência máxima, que é dada por:

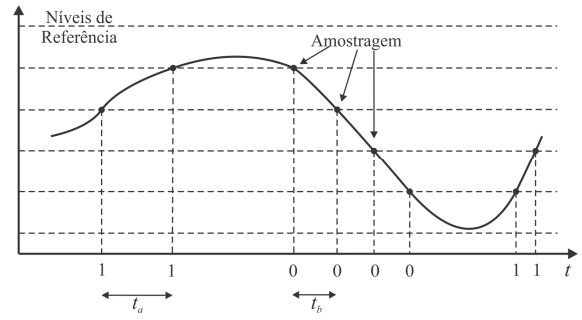


Figura 5. Sinalização a quatro fases.

$$f_{max} = \frac{1}{2\pi \cdot \delta_{max} \cdot 2^M} \quad (6)$$

2.3 Modulação Sigma-Delta Assíncrona

Os moduladores sigma-delta assíncronos apresentam níveis na saída correspondentes às subidas e descidas do sinal. Utilizando a amostragem por cruzamento de nível, gera-se "1" cada vez que o sinal cruzar para cima algum nível de referência ou "0" cada vez que o cruzamento seja para baixo, como se ilustra na Figura 5. Desta forma para mudanças rápidas e significativas do sinal, se gera uma sequência grande de bits. Caso contrário, se o sinal encontra-se em um estado quase estacionário, se gera uma quantidade reduzida de bits (Guan, 2006).

3 Projeto do ADC Assíncrono

O projeto do conversor análogo digital assíncrono apresentado neste trabalho é baseado na proposta de Filip Akopyan (Akopyan, 2006). Na Figura 6, se ilustra a arquitetura de divisores de tensão que originam os níveis de referência de comparação. Observa-se que para cada nível de referência tem-se um comparador analógico, um disparador digital e um bloco de processamento digital assíncrono que inclui o elemento de transmissão e armazenamento do token. Finalmente, o esquema inclui um bloco "XOR" que cria o fluxo de saída (tem níveis 1 ou 0 como em uma modulação

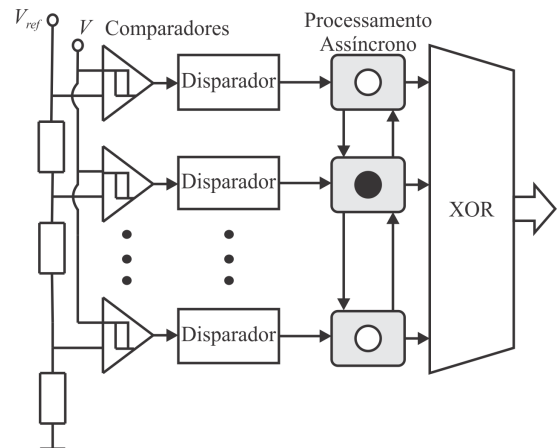


Figura 6. Arquitetura do ADC assíncrono.

delta sigma assíncrona via *dual rail*). Adicionalmente, pode-se incluir um contador de subida e descida para produzir os códigos binários.

3.1 Comparadores Analógicos

Na primeira etapa, um comparador analógico recebe o sinal e o compara com os níveis de referência, gerando um código de saída com 2^M-1 bits. Neste trabalho, o comparador usado foi construído com a tecnologia CMOS e os resistores do divisor de tensão foram implementados com o polisilício do circuito integrado visando-se baixo consumo de energia.

3.2 Disparador Digital

O disparador digital tem como objetivo o controle da comunicação entre o comparador e o elemento assíncrono, verificando se houve e não cruzamento. Caso esta verificação seja válida, passa-se à etapa seguinte ativando o bloco de processamento assíncrono e esperando sua resposta para iniciar nova etapa de controle.

Quando o disparador digital identifica a mudança na saída do comparador, verifica as variáveis de estado internas do elemento assíncrono que lhe corresponde, do anterior e do seguinte. Se todas as variáveis indicam que o cruzamento ocorreu, ele envia uma solicitação mediante um *handshake* a seu elemento correspondente de processamento assíncrono que em seguida gera uma saída, o disparador digital mantém a solicitação até que seja liberado mediante um sinal de habilitação para retornar a atender outra condição de cruzamento.

- **Primeiro Caso: Cruzamento Ascendente**

- Saída do comparador em nível alto 1;
- Elemento de processamento i indica que o sinal estava baixo;
- Elemento de processamento $(i-1)$ indica que o sinal estava alto;
- Elemento de processamento $(i-1)$ tinha completado sua tarefa: $ACK(i-1)=1$
- Elemento de processamento i tinha completado sua tarefa $ENABLE=1$;

- **Segundo Caso: Cruzamento Descendente**

- Saída do comparador em nível baixo 0;
- Elemento de processamento i indica que o sinal estava alto;
- Elemento de processamento $(i+1)$ indica que o sinal estava baixo;
- Elemento de processamento $(i+1)$ tinha completado sua tarefa: $DONE=1$
- Elemento de processamento i tinha completado sua tarefa $ENABLE=1$;

A solicitação é considerada atendida quando se produzir um dado na saída da unidade do disparador e em seguida ela for zerada automaticamente pelo sinal ACK do bloco de saída XOR, que se traduz em

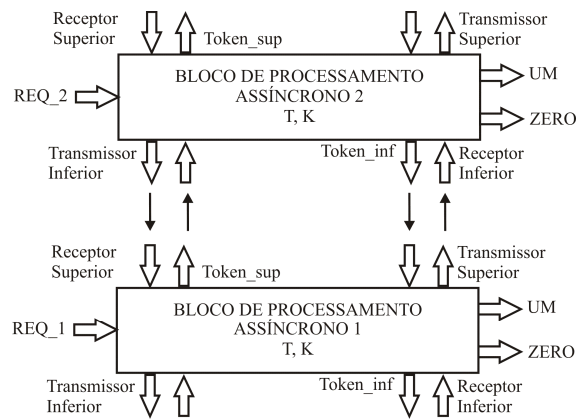


Figura 7. Blocos de processamento assíncrono.

um sinal $ENABLE$ quando chega ao bloco $DISPARADOR$.

3.3 Processamento Assíncrono

Para cada nível tem-se um bloco de processamento assíncrono, os quais se comunicam entre si, para poder enviar e receber um único token do sistema de acordo com o movimento do sinal de entrada, como se ilustra na Figura 7.

Desta maneira, a cada cruzamento de nível de referência um único token do sistema é transmitido a seu elemento digital assíncrono correspondente com o objetivo de dar exclusividade a operação. Assim, o elemento de processamento assíncrono controla as variáveis internas que indicam o sentido dos cruzamentos e produzem as saídas correspondentes para o elemento XOR. Os blocos de processamento assíncrono necessitam ter em seu elemento de memória o token para poder atender alguma solicitação do elemento $DISPARADOR$. No caso de que não tenha deverá solicitar a seus blocos vizinhos. Dessa forma, para as condições iniciais ou reset, um único token do circuito digital assíncrono encontra-se no primeiro nível inferior.

Cada elemento tem duas variáveis locais T e K , em que T indica a presença ou não do token no elemento e K tem duas interpretações, no primeiro caso se o bloco tem o token K indica o sentido do cruzamento do sinal e no segundo caso é quando o elemento não tem o token e K indica a localização do token, isto é, se o token está alto ou baixo.

Os processos que realizam o elemento de processamento assíncrono são:

- **Verificação do Token**

Este processo revisa se o elemento atual tem o $TOKEN$, se a resposta é verdadeira ($T=1$) termina o processo e é realizado o seguinte passo. Se o resultado não é verdadeiro ($T=0$), é necessário solicitar o token a um dos elementos assíncronos vizinhos, para o qual se baseia no valor de K :

- Se $K=1$ pede o token ao elemento superior
- Se $K=0$ pede o token ao elemento inferior.

Em seguida mediante um *handshake* com um de seus elementos vizinhos, o elemento atual recebe o token e fica preparada para continuar o processo.

- **Processo Principal**

Existem dois tipos de processamento principal, que dependem de qual das entradas foi ativada: se a solicitação vier do elemento DISPARADOR, passa-se ao processo de verificação de TOKEN e uma vez que o elemento possui o token, ele está habilitado para produzir a saída segundo o valor de K .

- Se $K=1$ a direção de cruzamento é descendente linha ZERO=1 e

- Se $K=0$ a direção de cruzamento é ascendente linha UM=1.

Os dados são enviados ao elemento XOR por um canal DUAL RAIL. Uma vez atendido, o elemento de processamento assíncrono muda o valor de K para seu complemento e envia o sinal ACK ao elemento DISPARADOR, indicando que está preparado para atender outra solicitação. Se a solicitação vier dos sinais de entrada “receptor superior” ou “receptor inferior”, significa que o elemento superior ou inferior está solicitando o token. O elemento realiza então o processo de verificação de TOKEN e uma vez obtido o token, ele é enviado mediante um *handshake*.

3.4 Bloco XOR

O bloco XOR produz os níveis para a etapa seguinte e o *handshake* para o canal DUAL RAIL. Neste estado, o sistema se encontra habilitado para atender outra solicitação de cruzamento de nível. Este bloco é formado basicamente por elementos XOR. Como passo final de implementação, em seguida ao bloco XOR, um contador binário ascendente e descendente, produz o código binário que pode ir acompanhado, segundo seja o caso, de um timer para uma possível reconstrução do sinal, como se ilustra na Figura 8.

4 Implementação do ADC Assíncrono em um FPGA STRATIX III/QUARTUS II

Nesta seção apresenta-se o procedimento de síntese do ADC assíncrono utilizando as ferramentas de projeto de linguagem de descrição de hardware padrão da IEEE. Foi usado VHDL, para assegurar uma completa portabilidade para qualquer plataforma de projeto convencional.

4.1 Síntese do ADC Assíncrono em um FPGA

A síntese e implementação do ADC Assíncrono baseou-se na arquitetura apresentada na seção anterior. Para este propósito, utilizou-se o QUARTUS II que é uma ferramenta de software utilizada para a simulação do projeto dos sistemas programáveis em chips (SOPC). Utilizou-se também o microwind 3.1, que é um software utilizado para a

simulação da parte analógica do circuito com a tecnologia de fabricação de circuitos integrados CMOS.

Para a implementação da parte digital do ADC Assíncrono foi utilizada a tecnologia FPGA STRATIX III, cujas características estão contidas na Tabela 1.

Tabela 1. Características do FPGA QUARTUS II

Características do FPGA	
Tecnologia	STRATIX III EP3SL50F484C2
Combinacional ALUTs	38000
Memória ALUTs	19000
Registradores	38000
Blocos DSP	384

Na Tabela 2 apresentam-se os resultados em recursos usados no FPGA para o projeto compilado para diferentes números de bits. Observa-se que se tem um baixo consumo de recursos para o projeto do ADC Assíncrono, isto permite disponibilizar os recursos excedentes para o projeto de outros blocos.

Tabela 2. Recursos utilizados no FPGA QUARTUS II

Características do Projeto		
Resolução (bits)	Quantidade de ALUTs no FPGA	Quantidade de registradores
3	403	7
4	936	15
5	1873	31
6	3782	63

4.2 Resultados de Simulação

Com a finalidade de realizar as simulações do processo de conversão do ADC assíncrono, definiu-se inicialmente o sinal de teste com características relacionadas a sinais quase estacionários, como se ilustra na Figura 9. Observa-se o funcionamento de cada nível de referência para diferentes números de bits.

As especificações para a simulação do projeto do ADC Assíncrono foi definida baseado no trabalho de Reanudin de laboratórios TIMA (Allier, 2005). (Tabela 3).

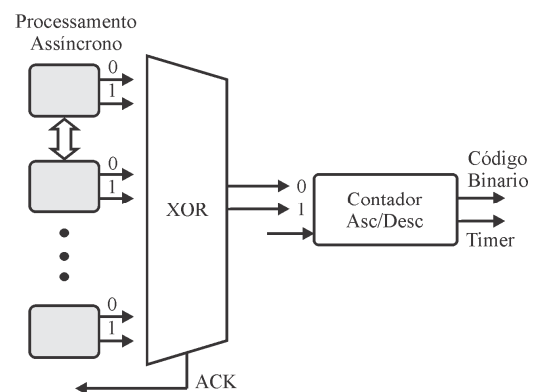


Figura 8. Bloco XOR e contador Ascendente e Descendente.

Tabela 3. Especificações para o projeto do ADC Assíncrono

Resolução (bits)	$M=4$ -bit
Tecnologia	180 nm CMOS STMicroelectronics
Tensão V_{dd}	1,8 v
Faixa dinâmica	$\Delta V = 0,8$ v
Retardo de processamento	$\delta = 93$ ns
Largura de Banda do sinal de entrada	$f_{max}=114$ kHz

A Tabela 4 contém os resultados que relacionam a potência consumida, a resolução em bits e a frequência máxima do sinal de entrada. Observa-se que quanto maior o número de níveis de referência maior a potência e menor a frequência, pois o sinal de entrada tem um número maior de cruzamento de níveis de referencia e tem um menor tempo para processar cada solicitação.

Tabela 4. Resolução em bits versus potência consumida.

Resolução (bits)	Potência (mW)	Frequência máxima da entrada
3	1,68	5 MHz
4	3,60	2,33 MHz
5	7,40	1,12 MHz
6	15,06	555,19 kHz

Na Tabela 5 mostram-se os resultados de simulação do ADC Assíncrono para uma resolução de 4 bits (15 níveis de cruzamento), no qual é realizado a análise das variações da largura de banda do sinal de entrada versus a potência consumida no processo de conversão. Estas frequências foram estudadas devido a que são incluídas em dispositivos sensores comerciais tais como sensores de pressão, temperatura, acelerômetros, entre outros. Os resultados obtidos são similares ao trabalho realizado por Akopyan para o projeto do conversor ADC assíncrono baseado em cruzamento de nível (Akopyan, 2006), verificando-se deste modo que a implementação utilizando a tecnologia FPGA Stratix III é adequada para este tipo de implementações. O projeto do conversor apresentado apresenta um tempo de conversão de 30 ns, que é próximo com os projetos dos conversores tipo FLASH.

Tabela 5. Largura de Banda versus Potência.

Largura de Banda do Sinal	Potência (μ W)
1 kHz	34,41
100 kHz	42,48
114 kHz	43,57
160 kHz	46,84
1 MHz	114,14
5 MHz	437,81

4 Conclusão

Neste trabalho foi apresentado o projeto de um conversor ADC Assíncrono implementado em uma tecnologia FPGA STRATIX III. A partir dos resultados de simulação verificou-se o desempenho do circuito projetado em CMOS, analisando-se as

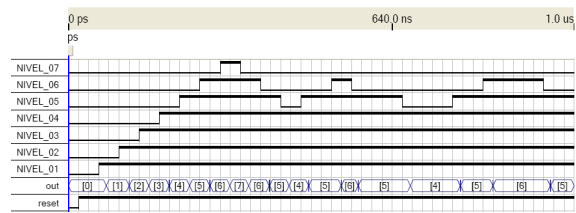


Figura 9. Sinal com características de quase estacionária.

relações entre a resolução em bits, a potência consumida no processo de conversão e a frequência máxima do sinal de entrada. Observou-se que à medida que a resolução aumenta, a potência consumida aumenta. Assim como também à medida que a frequência do sinal de entrada aumenta, a potência consumida também aumenta. Isto se deve principalmente, ao maior número de cruzamentos de níveis de referencia do sinal de entrada e um maior tempo de processamento. Conclui-se, pelos resultados de simulação, que o projeto baseado em uma tecnologia FPGA atingiu resultados similares a outros trabalhos realizados para este tipo de projetos.

Referências Bibliográficas

- Akopyan, F., Manohar, R. and Apsel, A. B. (2006) "A Level Crossing Flash Asynchronous Analog-to-Digital Converter", IEEE International Symposium on Asynchronous Circuits and Systems.
- Allier, E., Goulier, J., Sicard, G., Dezzani, A., André, E., Renaudin, M. (2005), "A 120nm Low Power Asynchronous ADC", IEEE International Symposium on Low Power Electronics and Design.
- Chinh, H. D. and Tan, Y. K. (2010), "Smart Wireless Sensor Networks", INTECH Open Access Publisher (www.intechopen.com último acesso 15/04/2012).
- Guan, K. and Singer, A. C. (2006) "A Level-Crossing Sampling Scheme for Bursty Signals", IEEE Conference on Information Sciences and Systems, pp: 1357-1359.
- Huot, N.; Dubreuil, H.; Fesquet, L.; Renaudin, M. (2005), "FPGA architecture for multi-style asynchronous logic [full-adder example]" Design, Automation and Test in Europe, pp: 32-33, Vol. 1.
- Matić, T.; Švedek, T.; Vinko, D. (2010) "Asynchronous Sigma-Delta Analog-to-Digital Converter", Circuits and Systems for Communications, pp: 111-114.
- Sparso, J. and Furber, S. (2001), "Principles of Asynchronous Circuit Design – A Systems Perspective". Kluwer Academic Publishers.
- Sutherland, I. E. (1989) "Micropipelines" Magazine Communications of the ACM, Vol 32, No6, pp.720-738.