

# PROJETO DE UM MANIPULADOR ROBÓTICO COM EXTREMIDADE DO TIPO GARRA ACIONADO POR FPGA

**Menotti Erasmo da Silva Machado<sup>1</sup>, menottismachado@gmail.com**  
**Francisco Helano Sampaio de Magalhães<sup>2</sup>, helano\_hsm@hotmail.com**  
**Natanael Rodrigues da Silva<sup>3</sup>, natanrs21@hotmail.com**  
**Pedro Hugo Menezes Rocha<sup>4</sup>, hugo\_mr19@hotmail.com**

<sup>1</sup>Universidade Federal do Ceará, Campus do PICI S/N, Bloco 725 Cx.Postal 6007, CEP. 60455-970. Fortaleza – CE,

<sup>2</sup>Universidade Federal do Ceará, Campus do PICI S/N, Bloco 725 Cx.Postal 6007, CEP. 60455-970. Fortaleza – CE,

<sup>3</sup>Universidade Federal do Ceará, Campus do PICI S/N, Bloco 725 Cx.Postal 6007, CEP. 60455-970. Fortaleza – CE,

<sup>4</sup>Universidade Federal do Ceará, Campus do PICI S/N, Bloco 725 Cx.Postal 6007, CEP. 60455-970. Fortaleza – CE,

## 1. INTRODUÇÃO

O emprego de sistemas digitais baseados em arranjos de portas lógicas programáveis por efeito de campo (FPGA – “Field Programmable Gate Array”) surgiu como uma alternativa viável para uma implementação mais flexível de sistemas de controle digital (Simard, 2009; Kariyappa, 2009). A característica intrínseca de funcionamento paralelo dos seus circuitos eletrônicos internos permite a realização de tarefas de forma simultânea, o que se mostra bastante adequado para o uso no acionamento de sistemas robóticos (Kongmunvattana and Congstitvatana, 1998). Esse funcionamento, eminentemente paralelo, possibilita o atendimento aos requisitos, de tempos de respostas críticos, do projeto de manipuladores robóticos (Huang, 2008). Segundo (Kongmunvattana and Congstitvatana, 1998), o emprego de FPGA’s no controle de robôs apresenta diversas vantagens em relação ao uso de computadores de propósito geral ou outros sistemas de microprocessados que possuem um único processador, principalmente quando as tarefas a serem desempenhadas são complexas. As funções lógicas de controle digital de robôs podem ser sintetizadas diretamente sobre os elementos lógicos do FPGA, possibilitando assim, a obtenção de estruturas mais simplificadas. O suporte a projeto de sistemas paralelos permite obter as tão desejadas respostas em tempo real. A baixa dissipação de potência desses circuitos, se comparado com um microprocessador ou até mesmo com um microcontrolador, possibilita uma economia considerável em termo de consumo de energia, o que permite uma maior autonomia aos sistemas que fazem uso daquela tecnologia.

A implementação do manipulador robótico tem como intuito o desenvolvimento de um sistema eletromecânico para auxiliar o ensino de técnicas de projeto lógico digital para alunos de engenharia de teleinformática e afins a partir de materiais reciclados de baixo custo. Na construção deste dispositivo, utilizaram-se motores de passo retirados de impressoras matriciais fora de uso, os quais podem ser facilmente obtidos em sucatas eletrônicas. Em sua estrutura de sustentação foi utilizado o acrílico, material leve, de baixo custo, resistente e de fácil manipulação.

A localização espacial do ponto terminal – garra, é representada através de um vetor de dimensão 4 com relação ao ponto de origem na base do manipulador, obtido através de uma translação sobre uma matriz de transformação linear que possui dimensão 4 x 4. Essa prática operação apresenta ao aluno uma forma útil de visualizar uma transformação linear, o que serve para sedimentar os conhecimentos adquiridos no domínio da álgebra linear, consolidando e fortalecendo assim, a intra-relação entre o conhecimento algébrico e a robótica.

Dado o seu baixo custo e fácil implementação, fatores que atenuam de modo contundente a característica interdisciplinar proposta, ele pode ser utilizado no ensino de álgebra vetorial e matricial nos estabelecimentos educacionais de ensino médio, representando uma ferramenta de inclusão social.

Este trabalho está organizado da seguinte forma: na Seção 2 denotam-se as características do protótipo desenvolvido; toda lógica de controle usada é descrita em Seção 3; na Seção 4, os resultados obtidos são evidenciados; na Seção 5 conclui-se o trabalho e apresenta-se as possíveis extensões futuras.

## 2. CARACTERIZAÇÃO E ESPECIFICAÇÕES DO PROTÓTIPO DESENVOLVIDO

### 2.1. Características Mecânicas

O modelo do manipulador robótico desenvolvido neste trabalho possui dois graus de liberdade, realizando os movimentos “pitch” ou arfagem e “yaw” ou guinada. A região do espaço ao alcance do manipulador construído é uma superfície semi-esférica, como mostrada na Fig. (1), sendo também definida pela Eq. (1).



Figura 1. Área esférica de atuação do manipulador

$$A = \int_{\varphi_1}^{\varphi_2} \int_{\theta_1}^{\theta_2} r^2 \sin\theta d\theta d\varphi \quad (1)$$

Aplicando os dados da Tab. (1) na Eq. (2) obtemos a área de trabalho do manipulador robótico mostrada na Fig. (1). O manipulador apresenta uma garra ou órgão terminal do tipo angular, classificação referente ao seu tipo de movimentação na realização de uma atividade qualquer. No efetuador do manipulador robótico, também designado “End of Arm Tool” - EOAT, o movimento angular torna-se peculiar, já que uma das hastes de fechamento apresenta-se fixa enquanto a outra haste permite movimentos tais como segurar e manipular objetos.

Tabela 1 . Indicação das variáveis matemáticas, bem como descrição dos valores que podem ser assumidos no decorrer do movimento

Variáveis	Valores assumidos
$r$	$2 \times 10^{-1} m$
$\theta$	$70^\circ - 130^\circ$
$\varphi$	$0^\circ - 360^\circ$

$$A = r^2 (\cos \theta_1 - \cos \theta_2) (\varphi_2 - \varphi_1) \quad (2)$$

O EOAT pode ser identificado como um ponto no espaço e representado através de uma matriz de dimensão 4x1, apresentada na Eq. (3), onde  $r$ ,  $\theta$  e  $\varphi$  representam os eixos clássicos do sistema de coordenadas esféricas e  $\omega$  um fator de escala real não-nulo.

$$\mathbf{u} = \begin{bmatrix} r \\ \theta \\ \varphi \\ \omega \end{bmatrix} \quad (3)$$

Utilizando-se a matriz de dimensão 4x4 definida em Eq.(4), responsável por uma transformação homogênea genérica, o deslocamento através da superfície de trabalho, pode ser modelado através da relação dada pela Eq. (5), onde  $\Delta\theta$  e  $\Delta\varphi$  representam, respectivamente, variações angulares nas coordenadas esféricas  $\theta$  e  $\varphi$ .

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta\theta \\ 0 & 0 & 1 & \Delta\varphi \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

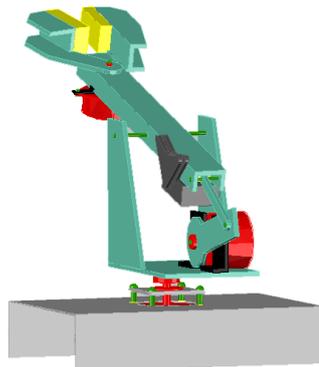
$$\mathbf{T} \cdot \mathbf{u} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta\theta \\ 0 & 0 & 1 & \Delta\varphi \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r \\ \theta \\ \varphi \\ \omega \end{bmatrix} = \begin{bmatrix} r \\ \theta + \Delta\theta \\ \varphi + \Delta\varphi \\ \omega \end{bmatrix} \quad (5)$$

Outra característica é a implementação de um sistema de atuação referente ao órgão terminal que possibilite a otimização da essência da garra: conversão da força oriunda dos motores para, através do eixo do mesmo, implantar uma força eficiente de outra natureza que resulte satisfatoriamente na realização da tarefa.

A estrutura do manipulador pode ser dividida em três partes principais: base, braço e garra. Em cada uma delas existe um motor de passo acoplado, responsável pela sua movimentação.

Para a construção do manipulador optou-se pelas juntas dos tipos rotacional e torcional. As juntas do tipo rotacional permitem uma maior velocidade escalar devido à presença de um eixo horizontal, que é ideal para os movimentos rápidos da garra e do braço. Já as juntas do tipo torcional, que podem se apoiar em superfícies, possibilitam um melhor desempenho para a sustentação de estruturas pesadas.

O manipulador robótico projetado e construído para a utilização em aulas práticas na disciplina de PLD apresenta a geometria observada na Fig.(2).



**Figura 2. O manipulador robótico**

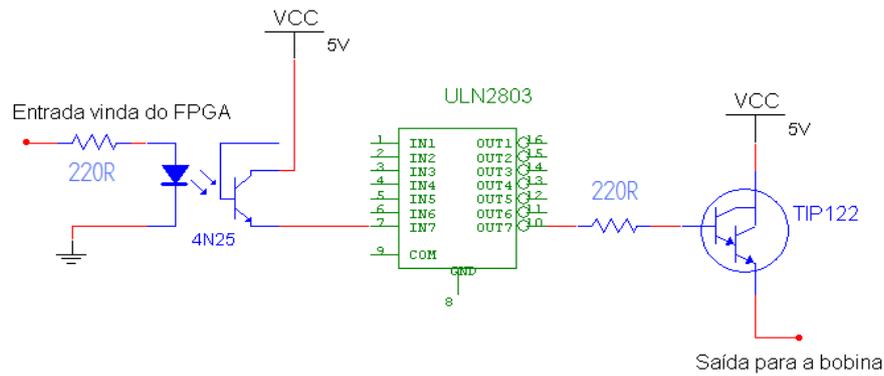
## 2.2. Características Elétricas

O acionamento das partes móveis é realizado por três motores de passo do tipo bipolar. Este tipo de motor foi escolhido por apresentar uma alta precisão de movimento em malha aberta, ter fácil implementação em aplicações digitais, além de não necessitar comutadores mecânicos (Olsson, 1992). As características elétricas de cada um dos motores encontram-se descritas na Tab. (2).

Como as correntes elétricas exigidas na operação dos motores são elevadas se comparadas àquelas fornecidas pelos pinos de saída do circuito FPGA, tornou-se necessário o projeto de circuitos condicionadores de sinais. Isso foi obtido através do uso de “drivers” de potência opticamente isolados.

**Tabela 2. Características elétricas dos motores de passo**

MOTORES	TENSÃO(V)	CORRENTE (mA)	Relação passo por ângulo
GARRA	12,0	0,5	7,5
BRAÇO	12,0	0,5	7,5
BASE	3,2	1,2	7,5



**Figura 3. Condicionador de sinal para o acionamento dos motores de passo**

Como se pode observar pela Fig. (3), o isolamento óptico é obtido com o emprego do fotoacoplador 4N25 (Boylestad, 1994). A corrente demandada pelos motores provém da amplificação da corrente que flui deste fotoacoplador, realizada pelo circuito integrado ULN2803 e pelo transistor TIP122. Utilizaram-se os dois arranjos em conjunto porque, conforme nos informa a Tab. (2), o consumo de corrente dos motores varia entre 0,5 e 1,2 mA, sendo que o ULN2803 podem fornecer no máximo 500 mA. Assim, por precaução, utilizou-se também o TIP122, para fornecer um ganho extra de corrente.

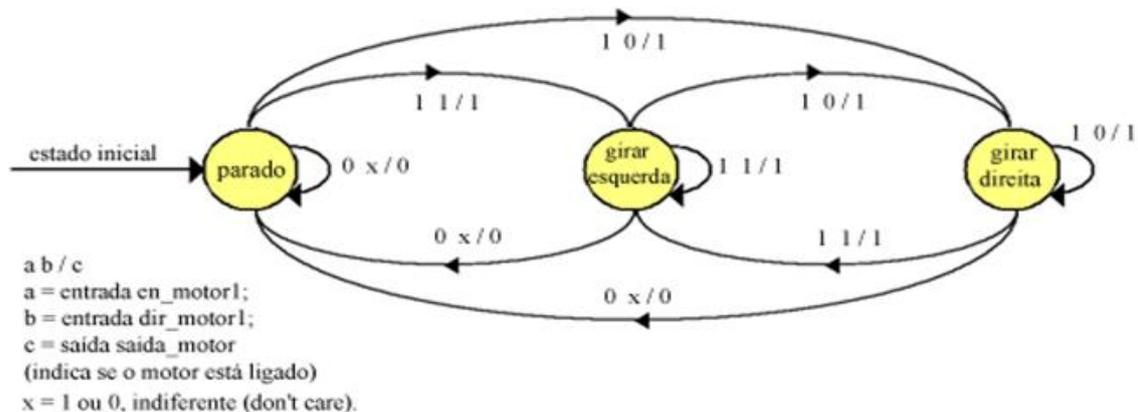
### 3. LÓGICA DE CONTROLE DIGITAL

Para o controle dos motores foi adotado um sistema controle em malha aberta, por considerar: (i) o torque exigido de cada um dos motores não sofre grandes variações; (ii) as posições dos motores são bem definidas pelos pulsos (Olsson, 1992), (iii) as operações são feitas a baixas velocidades. Nessas condições, um sistema desse tipo é mais indicado por requerer um algoritmo de controle de menor complexidade (Ngoc and Jae, 2007). Para implementar a lógica necessária para o correto acionamento dos motores, foi empregado o FPGA Cyclone II EP2C35F672C6 da Altera, presente no kit educacional DE2 (Altera, 2006).

Três estratégias foram elaboradas para o acionamento dos motores. Tais métodos diferem entre si, pelo tipo de acionamento requerido, a saber: (1) manual, (2) entrada numérica e (3) seqüencial de seqüência fixa. Desse modo, ocorre um aumento gradual do nível de dificuldade na implementação do código e nos testes com protótipo.

#### 3.1. Estratégia 1: Acionamento do Tipo Manual

O acionamento manual exige que o operador selecione os movimentos desejados a cada passo (Alves, 1988). O acionamento dos motores é realizado através de seis chaves do tipo *on-off* que definem o sentido de rotação (horário ou anti-horário) e o estado (habilitado ou desabilitado), de cada motor. Após efetuada a seleção do sentido, o operador então habilita o motor correspondente para a execução do movimento especificado até que seja atingido o alvo, sendo em seguida desabilitado. A figura (4) ilustra o diagrama de transição de estados para estratégia cujo pseudocódigo é apresentado em Anexo A.



**Figura 4. Diagrama de estados para a estratégia manual**

### 3.2. Estratégia 2: Acionamento por Entrada Numérica

O operador define, através de chaves, o valor de três variáveis de entrada: sentido (horário ou anti-horário), estado (habilitado ou desabilitado), e quantidade de passos (vetor binário). Em seguida, o operador, através de um pulso gerado na chave de estado, habilita o movimento. Após ser atingido o número de passos solicitado, os motores são automaticamente desenergizados, não sendo, necessária a atuação do operador. Anexo B apresenta o pseudocódigo para o acionamento por entrada numérica e a Fig. (5) ilustra o diagrama de transição de estados para essa estratégia.

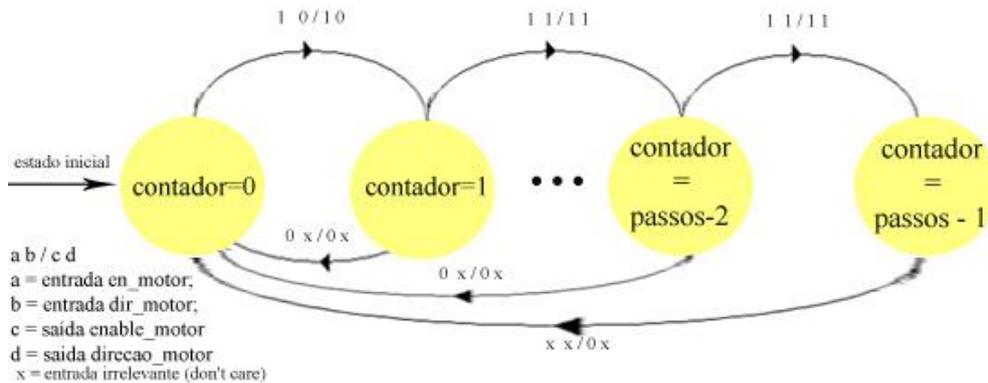


Figura 5. Diagrama de estados para a Estratégia Numérica.

### 3.3. Estratégia 3: Acionamento Sequencial de Seqüência Fixa

A terceira abordagem, a mais complexa, utiliza uma série de movimentos seqüenciais independentes da atuação de um agente externo, exceto o de fornecer o comando de inicialização da seqüência programada. Tal comando é fornecido pelo operador através de uma única variável de entrada. O movimento é dividido em etapas, onde cada um dos motores é acionado por vez, de forma que, ao término de cada etapa, a seguinte é ativada, não sendo mais necessária a atuação do operador. Figura (6) ilustra o diagrama de transição de estado para esta estratégia e o Anexo C apresenta seu pseudocódigo.

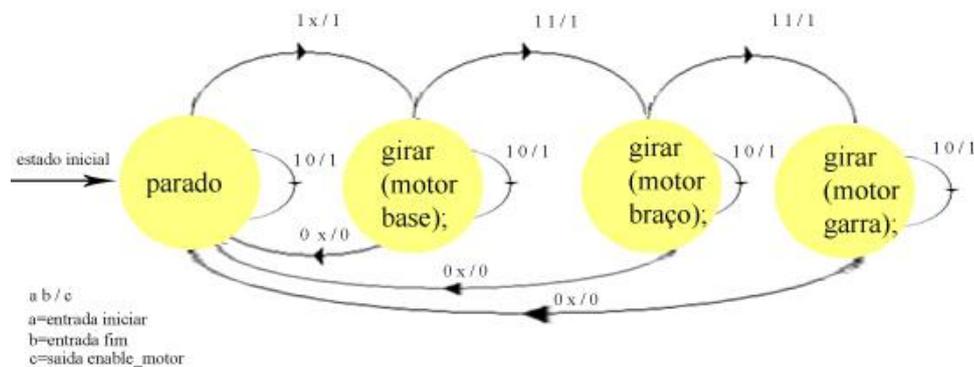


Figura 6. Diagrama de estados para a Estratégia Sequencial Fixa

## 4. RESULTADOS

A fim de avaliar o desempenho das estratégias de controle do manipulador robótico e devido à indisponibilidade de instrumentos de medições precisos no momento em que esse trabalho estava sendo feito, foram realizadas simulações no software Quartus II (Altera, 2007) para testar a validade dos resultados. Apresentam-se abaixo os resultados obtidos para cada uma das estratégias.

Para o acionamento manual, foram utilizados as seguintes variáveis:

*clock*: sinal de sincronização para a operação do sistema;

*dir\_motor1*: bit que representa a escolha do usuário do sentido de rotação do motor. Quando este sinal está em nível lógico alto, o motor está habilitado para girar da direita para a esquerda; quando está em nível lógico baixo, o motor gira no sentido oposto.

*en\_motor1*: sinal que aciona de fato o motor. Enquanto esse sinal é mantido em alto, o motor é livre para girar de acordo com o sinal *dir\_motor1*. Quando esse sinal está em baixo, o motor cessa seu movimento.

*seq\_motor1*: vetor de 4 bits que representa as bobinas do motor. Essa transição só ocorre se o sinal *em\_motor1* está em alto. A Tab. (3) mostrada a seguir, faz referencia a todas as variáveis utilizadas nos pseudocódigos anteriormente evidenciados.

**Tabela 3. Variáveis utilizadas nas estratégias de acionamento do manipulador robótico**

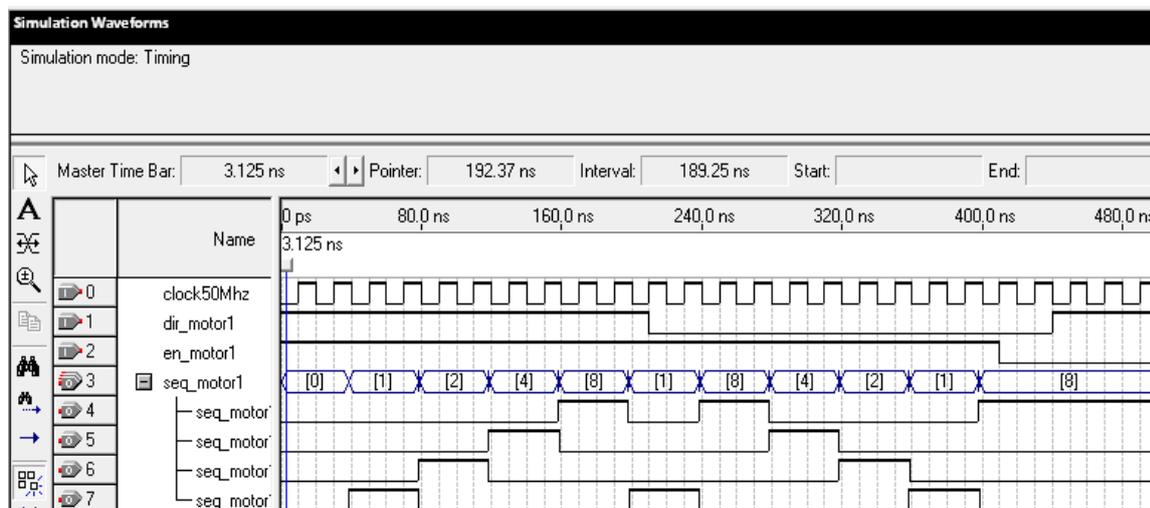
Variáveis	Descrição	Tamanho (bits)	Estratégias
clock:	Sinal de sincronização para a operação do sistema 50 Mhz	1	1, 2, 3
dir_motor1	Bit que representa a escolha do usuário do sentido de rotação do motor	1	1
en_motor1	Sinal que aciona de fato o motor.	1	1, 2, 3
seq_motor1:	Vetor de bits que representa a alternância das bobinas, caracterizando a existência e execução dos movimentos no motor.	4	1, 2, 3
passos:	Vetor de bits que recebe do usuário o número de passos(codificado em binário) que o motor deve executar;	4	2 e 3
fim:	Bit responsável por indicar o término do processo de movimento	1	2,3
iniciar:	Bit que apresenta a função de controlar e possibilitar movimento nos motores do manipulador.	1	3
enable_motor	Bit responsável por indicar a realização efetiva de movimento por parte dos motores.	1	1,2,3
direção_motor	Bit indicador da direção na qual o movimento é efetuado.	1	2

Tabela (4) enfatiza a função utilizada no pseudocódigo para a execução de determinada tarefa, auxiliando principalmente o entendimento por parte do leitor.

**Tabela 4 . Funções das estratégias de acionamento**

Funções	Descrição	Tipo parâmetro	Estratégias
Girar (parâmetro)	Função que retoma e executa o procedimento designado na estratégia 2, cada vez que é chamada .	inteiro	3

Figura (7) apresenta os resultados obtidos em simulação para a estratégia manual. O vetor de bits *seq\_motor1* representa a seqüência que comanda o movimento do motor. Observa-se que quando há uma inversão no sinal *dir\_motor1*, no instante 210 ns, ocorre também uma inversão na seqüência de passos gerada, fazendo com que o motor gire no sentido oposto. Observa-se também que quando o sinal *en\_motor1* vai para nível lógico baixo no instante 440 ns, não ocorrem mais transições nos sinais de acionamento das bobinas do motor, indicando que o sistema foi desabilitado pela ação do operador.



**Figura 7. Formas de onda para a estratégia manual**

Na estratégia de acionamento numérico, apresentada na Fig.(8), foram utilizadas as seguintes variáveis:

- clock*: sinal de sincronização para a operação do sistema;
- dir\_motor*: bit que representa a escolha do usuário do sentido de rotação do motor. Quando este sinal está em alto, significa que o motor deve girar para a esquerda; quando está em baixo, o motor deve girar para a direita.
- passos*: vetor de bits que recebe do usuário o número de passos(codificado em binário) que o motor deve executar;
- en\_motor*: sinal que recebe o indicativo do usuário de que a seqüência de passos deve ser executada. Quando ocorre um pulso nesse sinal, é dado início ao movimento do motor;
- seq\_motor*: vetor de 4 bits que representa as bobinas do motor.
- fim*: sinal de saída que apresenta um pulso toda vez que o motor termina de executar o número de passos determinado pelo usuário.

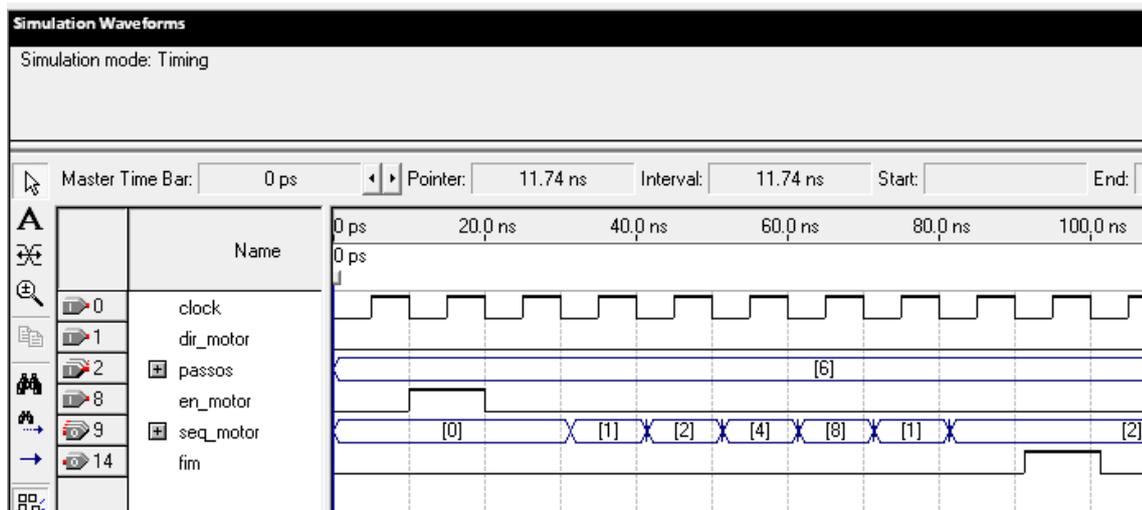


Figura 8 . Formas de onda para a estratégia numérica

Observe que a quantidade de passos desejada pelo operador (neste caso, seis) é selecionada através do vetor de bits *passos*, na terceira linha do diagrama. Uma transição positiva na entrada *en\_motor*, observada no instante 10 ns, habilita o início do movimento. Quando o movimento se completa, um pulso é gerado na saída *fim*, no instante 90 ns, indicando que a meta estipulada foi atingida.

Para a simulação da estratégia *seqüencial*, a mais complexa, admite-se uma seqüência autônoma de movimentos a fim de agarrar um objeto localizado em uma posição genérica  $r_0, \theta_0, \varphi_0$ , com a garra completamente aberta, pode ser observada na Fig. (9). Para tanto foram empregados as seguintes variáveis.

- clock*: sinal de sincronização para a operação do sistema;
- iniciar*: recebe a indicação de início do movimento. Quando o usuário provoca uma transição nesse sinal, a seqüência de movimentos é executada;
- motor\_base*: representa as bobinas do motor da base;
- motor\_braço*: representa as bobinas do motor do braço;
- motor\_garra*: representa as bobinas do motor da garra;

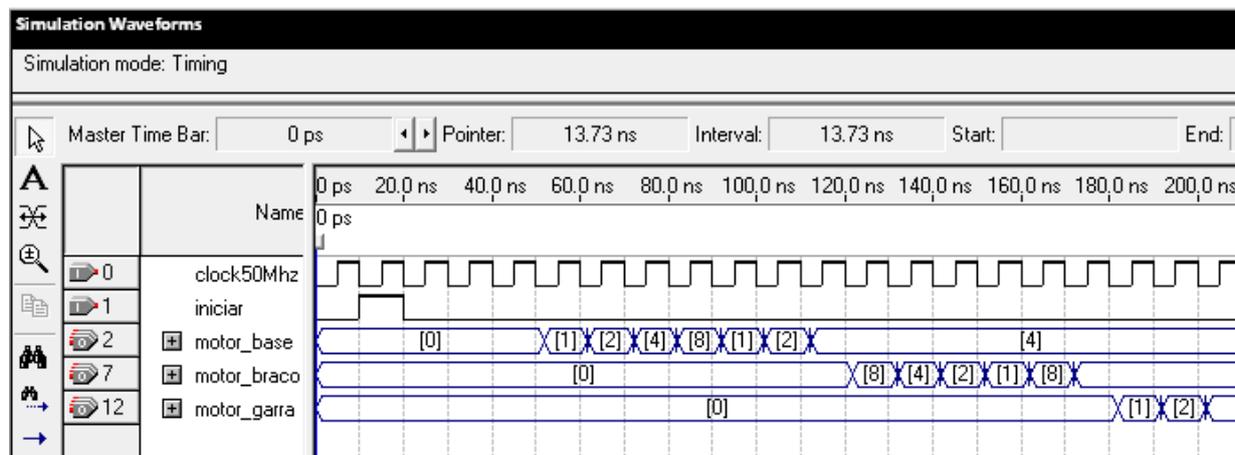


Figura 9. Formas de onda para a estratégia seqüencial fixa

A seqüência de movimentos executada é a seguinte: (i) o motor da base gira no sentido horário, a fim de posicionar o braço acima do objeto, como pode ser observado pela seqüência 0124812 da variável *motor\_base*, entre os instantes 50 e 110 ns; (ii) o motor do braço executa a seqüência 84218 na variável *motor\_braço*, fazendo o braço descer até o ponto desejado, como pode ser observado entre os instantes 120 e 170 ns; (iii) por último, o motor da garra é acionado, realizando a seqüência 124 sobre a variável *motor\_garra*, fazendo-o girar no sentido horário, como pode ser observado entre os instantes 180 e 200 ns. Observa-se que no instante 10 ns, a transição positiva na variável iniciar habilita o início da seqüência de movimentos.

A utilização do *clock* com frequência característica de 50 MHz, decorrente do FPGA, mostrou-se inviável para a obtenção dos resultados referentes às simulações lógicas, já que seria necessário a realização de um grande número de transições lógicas do sinal de controle para que a visualização do resultado condicional no algoritmo que está sendo testado seja compreendida. Para contornar essa problemática, optou-se por modificar o valor do número de transições necessárias à simulação, a fim de viabilizar tal processo.

É necessário ressaltar que quando esse procedimento de redução da referência relativa à simulação é tomado, torna-se inútil a intenção de exercer qualquer tipo de atividade, ou simplesmente movimento, com o manipulador robótico, devido à existência de uma incompatibilidade no valor e nas características dos sinais que forem designados anteriormente.

## 5. CONCLUSÕES E TRABALHOS FUTUROS

Baseado nos resultados obtidos, conclui-se que o emprego de FPGA para prototipagem apresenta a vantagem de permitir a mudança do algoritmo de controle com relativa facilidade, além de permitir a correção de erros com grande flexibilidade.

O processo de projeto e desenvolvimento do manipulador é uma ferramenta de ensino didática, pois engloba conhecimentos de diversas áreas, tais como álgebra linear, eletrônica analógica e digital, mecânica e lógica de programação.

As três estratégias foram facilmente implementadas em FPGA, sendo que cada uma apresenta vantagens em situações específicas. Das estratégias implementadas, a estratégia 1 é a que melhor se aplica a situações que exigem o domínio total dos movimentos por parte do operador em tempo real, para tomada de decisões. Ela se adequa perfeitamente para situações onde é necessária a intervenção remota do operador. A estratégia 3 pode ser utilizada onde não é imprescindível a presença de um operador. Em geral a seqüência foi definida com base nas ações que seriam tomadas por um operador para realização de um trabalho. Em geral seriam ações repetitivas e enfadonhas para em operador, como, por exemplo, a operação de robôs em linhas de montagem. Já a estratégia 2 representa um meio termo entre as estratégias 1 e 3. Quando ela é empregada, só há autonomia de deslocamento em cada um dos planos do movimento por vez. Ela é adequada quando a tarefa a ser desempenhada é dividida em sub-tarefas, e o início de cada uma destas sub-tarefas requer a atuação de um operador.

A fim de dar continuidade a este trabalho, objetiva-se implementar o controle em malha fechada, através da utilização de sensores de efeito Hall localizados nos ramos elétricos alimentadores de cada uma das bobinas dos motores de passo. Objetiva-se com isso identificar o comportamento das correntes de alimentação das bobinas quando ocorre à perda de passo e dessa forma poder corrigir o posicionamento do referido motor. A idéia básica reside no condicionamento dos valores das correntes para fins de aquisição através de conversores analógico-digital.

Dentre outras futuras melhorias que podem ser aplicadas a esse manipulador, destaca-se a busca por um aumento na quantidade de estratégias referentes a sua movimentação. Isso é necessário e interessante, já que possibilita um aperfeiçoamento e um aprimoramento perante as mais diversas tarefas que esse projeto robótico pode realizar quando aplicado à sociedade.

## 6. REFERÊNCIAS

- Altera Corporation, Altera DE2 Board, de2 Development and Educations Board – User Manual, 2006.
- Altera, Quartus II Version 7.2 Handbook, Altera Inc., Volume I, 2007.
- Alves, João Bosco Mota, “Controle de Robô”, Cartgraf, 1988.
- Boylestad. R. Dispositivos Eletrônicos e Teoria de Circuitos. 5Edição. Prentice Hall, 1994.
- Huang J. B. , Xie Z.W. , Liu · H. K. Sun · Y. C. Liu · Z. N. Jiang, DSP/FPGA-based Controller Architecture for Flexible Joint Robot with Enhanced Impedance Performance, Journal Intell Robot Syst (2008) 53:247–261.
- Kariyappa B. S., Hariprasad S. A., and R. Nagaraj, Position Control of an AC Servo Motor Using VHDL & FPGA, Proceedings of World Academy of Science, Engineering and Technology vol. 37, jan 2009.
- Kongmunvattana A. and Congstitvatana P., “A FPGA-based behavioral control system for a mobile robot,” in *Proc. IEEE Asia-Pacific Conf. Circuits and Systems*, 1998, pp. 759-762.
- Malvino, Albert Paul, “Princípios de Eletrônica”, Makron books, 1997.
- Ngoc Quy Le and Jae Wook Jeon, An Open-loop Stepper Motor Driver Based on FPGA, International Conference on Control Automation and System 2007 in COEX, Seoul, Korea
- Olsson G., Piani G., “Computer Systems for Automation and Control”, Prentice Hall International, 1992.
- Simard S., Mailloux, J.-G. and R. Beguenane, Prototyping Advanced Control Systems on FPGA, EURASIP Journal on Embedded Systems, Mars 03 2009.

## 7. ANEXOS

### Anexo A – Pseudocódigo para a estratégia manual

```
declaração de variáveis
clock, dir_motor1, en_motor1: in bit;
enable_motor: out bit;
seq_motor1: vetor de bit;
inicio
  repita
    se clock = '1' então
      se en_motor1 = '1' então
        enable_motor='1';
      se dir_motor1 = '1' então
        girar_para_esquerda(seq_motor1);
      senão
        girar_para_direita(seq_motor1);
      fim se;
    fim se;
  fim se;
fim repita;
fim
```

### Anexo B – Pseudocódigo para a estratégia numérica

```
declaração de variáveis
passos, contador: inteiro;
seq_motor: vetor de bit;
clock, en_motor, fim, dir_motor: in bit;
direcao_motor, enable_motor: out bit;
inicio
  repita
    se clock = '1' então
      se en_motor = '1' então
        se contador < passos então
          contador = contador + 1;
          enable_motor='1';
          se dir_motor = '1' então
            direção_motor='1';
            girar_para_esquerda(seq_motor1);
          senão
            enable_motor='1';
            direção_motor='0';
            girar_para_direita(seq_motor);
          fim se;
        senão
          contador=0;
          enable_motor = '0';
          fim='1';
        fim se;
      fim se;
    fim se;
  fim repita;
fim
```

## Anexo C – Pseudocódigo para a estratégia seqüencial fixa

```
declaração de variáveis
motor_base, motor_garra, motor_braco: vetor de bit;
iniciar, fim, clock: in bit;
passos_base, passos_braco, passos_garra: inteiro;
enable_motor: out bit;
inicio
  repita
    se clock = '1' então
      se iniciar = '1' então
        enable_motor='1';
        girar(passos_base);
        girar(passos_braco);
        girar(passos_garra);
      fim se;
    fim se;
  fim repita;
fim
```

## DESIGN OF A MANIPULATOR WITH END TYPE GRAB POWERED BY FPGA

Menotti Erasmo da Silva Machado<sup>1</sup>,  
Francisco Helano Sampaio de Magalhães<sup>2</sup>, helano\_hsm@hotmail.com  
Natanael Rodrigues da Silva<sup>3</sup>,  
Pedro Hugo Menezes Rocha<sup>4</sup>,

<sup>1</sup>Federal University of Ceará, Campus PICI innumerable, Block 725 Mailbox 6007, CEP. 60455-970. Fortaleza – CE,

<sup>2</sup>Federal University of Ceará, Campus PICI innumerable, Block 725 Mailbox 6007, CEP. 60455-970. Fortaleza – CE,

<sup>3</sup>Federal University of Ceará, Campus PICI innumerable, Block 725 Mailbox 6007, CEP. 60455-970. Fortaleza – CE,

<sup>4</sup>Federal University of Ceará, Campus PICI innumerable, Block 725 Mailbox 6007, CEP. 60455-970. Fortaleza – CE,

**Abstract:** *This work focuses on the design and control of a robotic manipulator using FPGA (Field Programmable Gate Array) circuits. Three open-loop control strategies were developed to move the manipulator, through the activation of three bipolar stepper motors. The first strategy considers a manual control; the second one consists of a numerical control, and the third one uses fixed sequence control inputs.*

*The manipulator structure is organized in three main parts: base, arm and claw. Each one of them with a stepper motor connected, making possible pitch and/or yaw movements. The manipulator is mechanically constituted by rotational and torsional joints, and its work space corresponds to a semi-spherical surface.*

*The programming and simulation of the logical expressions used in the control algorithms were coded in VHDL (VHSIC Hardware Description Language), using the Altera Quartus II software. The generated code was synthesized on Cyclone II FPGA, which is part of the Altera DE2 development kit.*

*The computer simulation results showed that digital control circuits correctly perform the scheduled tasks for all three tested control strategies. The developed robotic arm is able to execute three different and complementary movements: roll motion in the base, another to alight the arm that holds the claw, and a third to open or close it.*

*It is remarkable that the flexibility obtained by the use of FPGA devices allows modifications of control strategies without the need of interrupting the operation of the implemented robotic system, which is a desirable feature in an industrial environment. The prototype can be used in environments where the task to be performed poses risks to human health, such as handling toxic and/or radioactive materials.*

**Keywords:** *FPGA, VHDL, robotic manipulator, digital control, stepper motor.*