

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC	
(X) PRÉ-PROJETO () PROJETO	ANO/SEMESTRE: 2024/1

IMPLEMENTAÇÃO DE UM COMPUTADOR DE 8 BITS DENTRO DE UMA FPGA

Sézio Clener Nascimento

Prof. Miguel Alexandre Wisintainer – Orientador

1 INTRODUÇÃO

Raros são os produtos tecnológicos que evoluíram tão rapidamente quanto os computadores digitais. Há cerca de 40 anos, os computadores utilizavam válvulas, requeriam enormes quantidades de energia e instalações de ar condicionado. A evolução tecnológica permitiu que isso pudesse ser feito com poucos componentes integrados, consumindo energia equivalente a pouco mais do que um aparelho de televisão, e a princípio, sem a necessidade de um sistema de refrigeração (MONTEIRO, 1996).

Com isso, passou-se de computadores gigantes capazes de realizar algumas centenas de cálculos por segundo (VAUGHAN-NICHOLS, 2018), a minúsculos chips contendo bilhões de transistores e racks de servidores capazes de executar quadrilhões de operações por segundo e transferir dados a taxas gigantescas (CUTRESS, 2018). Atualmente, estes chips estão divididos em três categorias: *Application-Specific Standard Products* (ASSPs), *Application-Specific Integrated Circuits* (ASICs) e os *Field Programmable Gate Array* (FPGAs).

Os ASICs são componentes criados para uma única aplicação e são desenvolvidos para serem usados, normalmente, por apenas uma empresa. Os ASSPs são desenvolvidos para aplicações de uso geral, como os processadores dos nossos computadores, e são vendidos para uma quantidade maior de clientes. Mas os ASICs e ASSPs tem suas limitações. Para criá-los é necessária uma equipe de engenheiros capazes e de grandes recursos financeiros. Além de demandar um tempo de desenvolvimento chegando a 1 ano (MURALI, 2013).

Os FPGAs, segundo Moore e Wilson (2017), são dispositivos semicondutores, cuja função pode ser definida após sua fabricação, através de uma linguagem de descrição de hardware, como por exemplo: (*VHSIC Hardware Description Language*) VHDL, ou Verilog.

Com base no que foi dito, este trabalho propõe implementar um computador de 8 bits dentro de uma FPGA, a fim de demonstrar o funcionamento de um sistema computacional, proporcionando uma compreensão mais aprofundada do seu funcionamento.

2 OBJETIVOS

Objetivo geral deste trabalho será portar um computador de 8 bits para dentro de uma FPGA.

Os Objetivos específicos serão:

- carregar um programa pré-programado em uma memória externa;
- realizar a leitura de entradas, processá-las e devolver o resultado para as saídas;
- documentar de forma simples e detalhada o funcionamento do computador;
- compreender o funcionamento de sistemas computacionais mais complexos.

3 TRABALHOS CORRELATOS

Foram estudados quatro trabalhos correlatos. O primeiro descreve o funcionamento da máquina M++ dentro de uma FPGA (ANDRÉ LEONARDO BIEGING, 2018). O segundo é outra CPU também implementada em uma FPGA (NAYARA GUIMARÃES DUTRA, 2016). O terceiro trabalho foi um videogame desenvolvido dentro de uma FPGA (LAURA QUEVEDO JURGINA et al., 2019). O quarto trabalho foi um reproduzidor de imagem através de uma entrada VGA usando uma FPGA (MICHAEL ALEJANDRO DIAS et al., 2007).

3.1 IMPLEMENTAÇÃO DA MÁQUINA M++ EM FPGA

A máquina M++ dentro de uma FPGA foi desenvolvida pelo aluno André Leonardo Bieging (2018), do curso de Ciências da Computação da FURB. Ele portou o projeto da máquina M++ (desenvolvida por Borges, 2003), para dentro de uma FPGA.

Trabalho este que já tinha sido desenvolvido por outros alunos dentro do simulador Digital Works. No entanto, o trabalho do aluno André Leonardo Bieging (2018) foi desenvolvido como uma prova de conceito da aplicação do processador dentro de uma FPGA. O objetivo de seu trabalho foi implementar dentro de uma FPGA a arquitetura da máquina M++, criada na FURB e desenvolvida para funcionar apenas dentro de um ambiente de simulação. Seu trabalho provou que a mesma pode funcionar dentro de um chip real. O que motivou ele a fazer isso foi poder aumentar a performance da arquitetura da máquina M++, podendo chegar à casa dos MHz.

3.2 ESTUDO E IMPLEMENTAÇÃO EM FPGA DE UM MICROPROCESSADOR

Este trabalho foi desenvolvido pela aluna Nayara Guimarães Dutra (2016), do curso de Engenharia Elétrica, do Centro Federal de Educação Tecnológica de Minas Gerais. Nele, ela desenvolveu um microprocessador por meio de lógica programável dentro de uma FPGA.

Inicialmente, ela realizou um estudo a respeito dos microprocessadores, arquiteturas e das linguagens de descrição de hardware passíveis de utilização para a implementação do microprocessador de 8 bits, denominado de (*Simple As Possible*) SAP-1.

O computador SAP-1 possui uma arquitetura projetada para fins acadêmicos. Sua simplicidade e inteligibilidade podem ser alcançadas com um número reduzido de instruções e baixa velocidade de processamento. Apesar de ser o primeiro estágio na evolução dos computadores, este computador possui muitos conceitos capazes de introduzir e exemplificar os sistemas básico de um computador.

"O SAP-1 possuía um barramento de 8 bits por onde trafegavam os sinais para a execução das instruções do microprocessador. As saídas possuíam conexão com o barramento representadas por *tri-states*, ou seja, saídas de três estados, permitindo a geração de valores 0 (baixo), 1 (alto) ou Z (alta impedância). Por ser formado por um conjunto de portas lógicas, a análise do sistema era feita com auxílio de funções booleanas."

3.3 PROTOTIPAÇÃO EM FPGA DE UM VIDEO GAME DESENVOLVIDO EM LINGUAGEM VHDL

O projeto desenvolvido pelos alunos Laura Quevedo Jurgina, et al. (2019), consistiu em desenvolver um jogo simples e intuitivo dentro de uma FPGA. Este projeto reproduz o conceito de projetos anteriores, nos quais foram emulados outros jogos desenvolvidos para o Atari 2600 (MONTFORT; BOGOST, 2009) em uma placa FPGA (FLACH et al., 2012). O projeto foi completamente implementado em VHDL para ser executado no kit de desenvolvimento Altera DE2, equipado com o FPGA Cyclone II.

Para isso, foi utilizado a saída VGA fornecida pela placa para se comunicar com um monitor de vídeo. O jogo foi apresentado em um monitor de vídeo, com uma resolução de 1280 por 1024 pixels e com uma taxa de atualização de 60 Hz. A frequência necessária para a atualização do pixel na imagem era de 108 MHz. Contudo, o valor fornecido pela placa era de 27 MHz. Sendo assim, foi necessário construir um multiplicador de frequência para elevar a frequência original para frequência desejada.

3.4 REPRODUZINDO UMA IMAGEM EM UM MONITOR VGA USANDO UMA FPGA

Este projeto foi desenvolvido pelos alunos Michael Alejandro Dias, et al. (2007) e consistiu em desenvolver um aplicativo em Visual Basic que poderia capturar as informações de uma imagem para um periférico contendo uma FPGA. A FPGA processava estas informações da imagem e as enviava para um monitor através da saída VGA da própria FPGA. Para isso, foi utilizada a linguagem VHDL e a implementação foi realizada utilizando a placa FPGA Cyclone II desenvolvida pela própria Altera e a ferramenta de síntese foi o Quartus II.

Este artigo descreve todo o processo de como capturar uma imagem a partir do Visual Basic, enviá-la para uma FPGA, processá-la e transmiti-la para um monitor de vídeo através da entrada VGA. O módulo receptor UART consiste em dois blocos projetados em VHDL que foram unidos: um dos blocos é para definir a taxa de transmissão e o segundo módulo é uma máquina de estado que é responsável por armazenar os dados que chegam.

O Quadro 1 faz uma comparação entre os trabalhos correlatos citados anteriormente.

Quadro 1 – Comparativo dos trabalhos correlatos.

Correlatos Caract	André (2018)	Nayara (2016)	Laura et al. (2019)	Michael et al. (2007).
Fabricante	Altera	Altera	Altera	Altera
Tecnologia	FPGA	FPGA	FPGA	FPGA
Ferramenta	Quartus prime	Quartus II	Quartus II	Quartus II
Arquitetura	8 bits	8 bits	-	-
Linguagem	Verilog	VHDL	VHDL	VHDL

Fonte: O autor (2024).

Todos os trabalhos utilizaram os FPGAs do fabricante Altera. O primeiro trabalho correlato utilizou o Verilog como linguagem de descrição de hardware, enquanto os outros trabalhos correlatos optaram por usar o VHDL para a descrição de hardware. Este trabalho será desenvolvido utilizando também uma FPGA da Altera.

4 ARQUITETURA ATUAL

A arquitetura da máquina M++, dentro de um FPGA, foi desenvolvida pelo aluno André Leonardo Biegging (2018). A partir da arquitetura da máquina M++, criada na FURB para funcionar apenas dentro de um simulador, o objetivo era demonstrar que ela poderia funcionar dentro de um chip real. Outro motivo era que essa tecnologia poderia aumentar a performance da arquitetura utilizada pela máquina M++ anterior, podendo atingir a casa dos MHz, assim como os outros projetos correlatos. A arquitetura também utiliza uma FPGA para a sua concepção. A arquitetura da máquina M++ consistia de várias partes de grande importância em qualquer arquitetura de computador, como:

- a) unidade lógica e aritmética: é responsável por ler dados do acumulador e de algum registrador, e realizar operações lógicas e aritméticas com estes. As operações possíveis são: soma, subtração, AND, OR, XOR e NOT;
- b) acumulador: responsável por conter um dado de 8 bits temporariamente, até que este seja utilizado por algum outro bloco;
- c) registradores: contém uma célula memória de 8 bits. Esses valores podem ser utilizados para realizar operações na ULA ou podem ser mostrados nas saídas;
- d) barramento de dados: responsável por transferir os dados entre todos os blocos do microprocessador. Pode transferir até 8 bits simultaneamente;
- e) banco de Portas: utilizado para realizar operações de entradas e saídas. Pode ler entradas e acionar saídas;
- f) memória EEPROM: é uma memória de apenas leitura onde é armazenado o programa lido.

5 PROPOSTA DO MICROPROCESSADOR

É muito importante para a formação de um Cientista da Computação entender o funcionamento interno de um computador, para ser capaz de elaborar soluções mais eficientes. Uma das disciplinas fundamentais para isso é a Arquitetura de Computadores (VIANA et al., 2009).

Este projeto tem como proposta o desenvolvimento de um computador simples de 8 bits dentro de uma FPGA, que possa demonstrar, através de exemplos de código, como o computador realiza determinadas tarefas comuns do cotidiano. Trazendo assim um conhecimento mais profundo do funcionamento de um sistema micro processado, fazendo com que os alunos do curso de Ciências da Computação tenham uma ideia mais ampla do funcionamento de um sistema computacional. Este projeto já foi implementado por outros autores em hardware, como, por exemplo, o descrito no livro “digital computer electronics”. No entanto, este será implementado dentro de uma FPGA, usando, para isso, uma linguagem de descrição hardware.

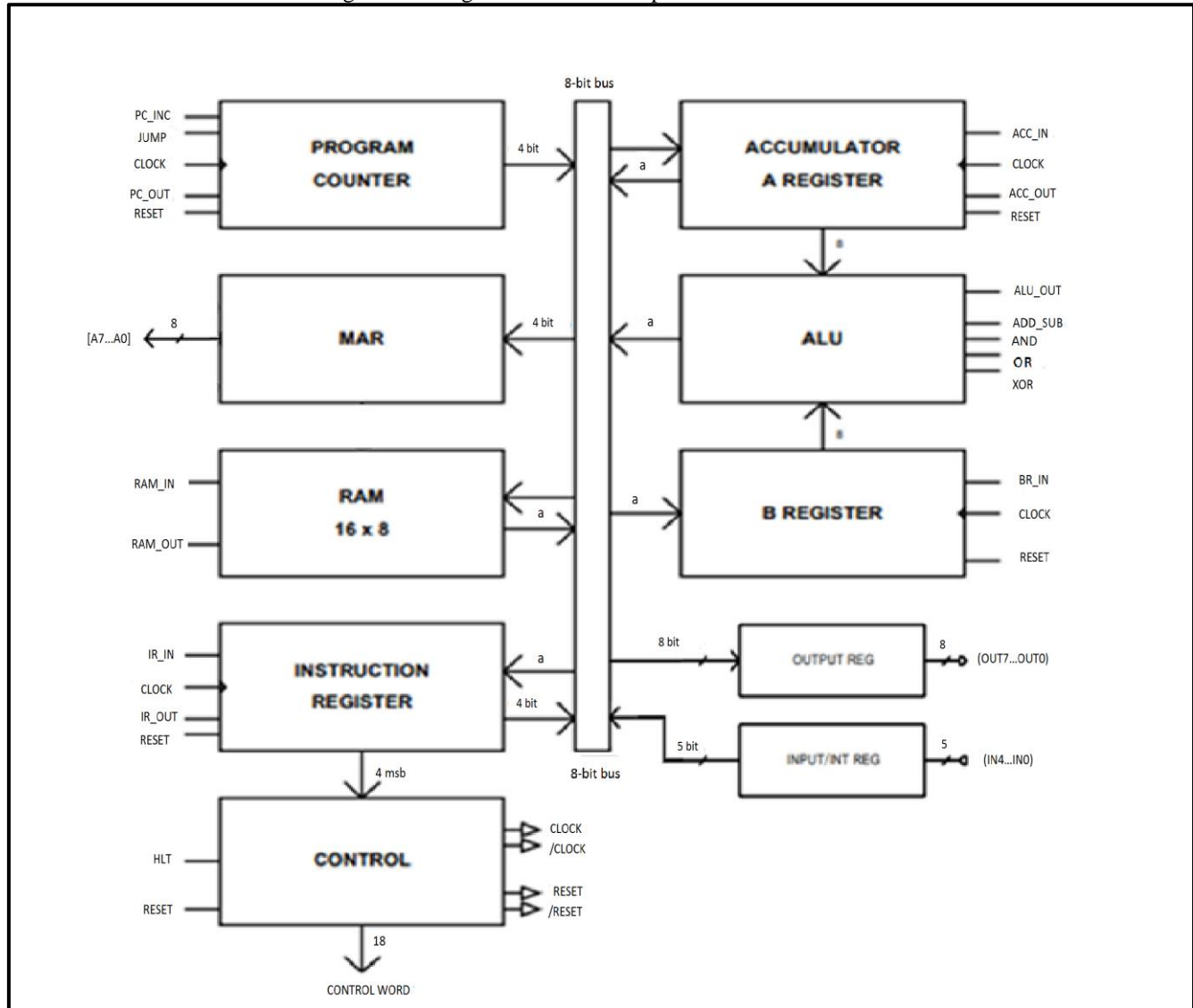
Cada bloco do computador apresentado pelos autores anteriores, e descrito na Figura 1, pode ser montado em hardware, usando para isso componentes físicos, como portas lógicas, Flip Flop, registradores. No entanto, este projeto será feito utilizando uma linguagem de descrição de hardware, neste caso, o Verilog. Sendo assim, cada bloco descrito abaixo será representado como um arquivo individual dentro de uma FPGA.

Todo código fonte será armazenado em uma memória externa, a qual não foi apresentada no diagrama de bloco do projeto devido a não fazer parte do hardware proposto pelo autor, assim como a memória de programa e

os periféricos de entrada e saída, como displays e botões, também foram omitidos do diagrama de bloco. Isso ocorre porque estes periféricos não são o foco do projeto, sendo apenas dispositivos usados para apresentar os resultados finais.

A representação abaixo é um diagrama em bloco do computador de 8 bits proposto pelo autor.

Figura 1 – Diagrama em bloco do processador.



Fonte: O autor (2024).

5.1 JUSTIFICATIVA

Este trabalho será desenvolvido com o propósito de criar um processador ou computador de 8 bits para ser um concorrente direto da máquina M++, que foi desenvolvida no *Logisim* e depois implementada em um FPGA pelo aluno André Leonardo Bieging (2018). Também, este projeto tem como objetivo fazer com que os alunos do curso de Ciência da Computação tenham mais interesse em estudar os fundamentos da computação, tanto na área de hardware quanto na área de software. A ideia principal do projeto é desenvolver um computador de 8 bits que suporte um conjunto de instrução reduzido. As instruções que o computador deve aceitar estão citadas no Quadro 2.

Quadro 2 – Instruções do processador.

Mnemônico	Descrição	Código de Operação	Ciclos de Clock
NOP	Sem operação: processa o ciclo de busca e não realiza qualquer movimentação de dados no ciclo de execução.	0000	5
LDA	Utilizada para mover um dado de 8 bits para o acumulador.	0001	5
LDI	Carrega acumulador por endereçamento imediato.	0010	5
STA	Armazena conteúdo do acumulador em um registrador de uso geral (uma das posições da memória RAM).	0011	5
ADD	Soma o conteúdo do acumulador com o conteúdo da posição de memória apontada pelo operando da instrução.	0100	5
SUB	Subtrai o conteúdo da posição de memória apontada pelo operando da instrução do acumulador.	0101	5
AND	Realiza a operação lógica AND bit a bit entre o conteúdo do acumulador e o conteúdo da posição de memória apontada pelo operando da instrução.	0110	5
ORL	Realiza a operação lógica OR bit a bit entre o conteúdo do acumulador e o conteúdo da posição de memória apontada pelo operando da instrução.	0111	5
XOR	Realiza a operação lógica XOR bit a bit entre o conteúdo do acumulador e o conteúdo da posição de memória apontada pelo operando da instrução.	1000	5
NOT	Realiza a operação lógica NOT bit a bit no acumulador. Salva o resultado no próprio acumulador.	1001	5
JMP	Realiza um desvio incondicional para o endereço de memória apontado pelo operando da instrução.	1010	5
OUT	Envia o conteúdo do acumulador para o registrador de entrada e saída.	1011	5
INPUT	Envia o conteúdo da saída para o registrador de entrada e saída.	1100	5
HLT	Para o processamento, desligamento do sinal de clock.	1101	5
PUSH	Empilha um elemento no topo da pilha.	1110	5
POP	Desempilha um elemento do topo da pilha	1111	5

Fonte: O autor (2024).

Cada operação do processador, descrito no Quadro 2, terá um número binário de quatro bits, que é o código da operação. Esse código representa os quatro bits mais significativos do byte, também conhecido como o *nibble* mais significativo. Junto com os quatro bits do contador de ciclo de instrução, que representa os quatro bits menos significativo do byte, também conhecido como o *nibble* menos significativo, formarão o endereço da palavra de controle da instrução resultante.

O endereço da palavra de controle, nada mais é do que o código completo da instrução. No computador de 8 bits, toda esta programação será feita em binário. Podendo, em projetos futuros, ser desenvolvido um compilador para que este computador seja programado diretamente em *assembly*.

O Quadro 3, representa o ciclo de busca do processador.

Quadro 3 – Ciclo inicial de todas instruções.

Ciclo	Origem do dado	Destino do dado
T0	PC	MAR
T1	RAM	<i>Instruction register</i>

Fonte: O autor (2024).

O contador de ciclo de instrução serve para contar as operações que devem ser realizadas dentro de uma única instrução. O Quadro 4 tem como propósito demonstrar o que será executado a cada ciclo de instrução do processador. Para simplificar a construção do projeto, as instruções do processador foram todas definidas em 5 ciclos de máquina, sendo que o primeiro e o segundo ciclo são os ciclos de busca (chamado aqui de Tciclo), já apresentados na tabela acima.

Quadro 4 – Sequência de registradores afetados.

Mnemônico	OPCODE	T2	T3	T5
NOP	0000	-	-	-
LDA	0001	IRO, MAI	RMO, ACI	-
LDI	0010	IRO, ACI	-	-
STA	0011	IRO, MAI	ACO, RMI	-
ADD	0100	IRO, MAI	RMO, BRI	ALU, ACI
SUB	0101	IRO, MAI	RMO, BRI	ALU, ACI, SUB
AND	0110	IRO, MAI	RMO, BRI	ALU, ACI, AL0
ORL	0111	IRO, MAI	RMO, BRI	ALU, ACI, AL1
XOR	1000	IRO, MAI	RMO, BRI	ALU, ACI, AL1, AL0
NOT	1001	ALU, ACI, AL1, AL0, NOT	-	-
JMP	1010	IRO, PCI	-	-
OUT	1011	ACO, ORI	-	-
INPUT	1100	ORI, ACO		
HLT	1101	HLT	-	-
PUSH	1110	PCO, MAI	PCI, IRO	-
POP	1111	MAO, PCI	-	-

Fonte: O autor (2024).

O Quadro 5 descreve as micro instruções que devem ser realizadas pelo processador a cada instrução. Cada micro instrução representa o acesso a um dos blocos do processador, sendo que esta poderá ser um acesso de entrada a um bloco ou um acesso de saída a um bloco.

Quadro 5 – Micro instruções com as respectivas descrições.

ACI	Acumulador é posto como entrada.
ALU	Unidade Lógica aritmética.
ACO	Acumulador é posto como saída.
BRI	Registrador B é posto como saída.
IRO	Registrador de Instrução é posto como saída.
MAI	Registrador de endereço de memória é posto como entrada.
MAO	Registrador de endereço de memória é posto como saída.
ORI	Acesso ao registrador de entrada e saída.
PC	Contador de programa.
PCI	Contador é posto como entrada.
PCO	Contador é posto como saída.
RMI	Registrador de memória é posto como entrada.
RMO	Registrador de memória é posto como saída.

Fonte: O autor (2024).

5.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Este trabalho tem como objetivo atender aos seguintes requisitos funcionais e não funcionais:

- a) armazenar o código objeto em uma memória externa (Requisito Funcional – RF);
- b) ler o programa armazenado em memória externa (RF);
- c) realizar a leitura de entradas (RF);
- d) apresentar os dados em suas saídas (RF);
- e) a memória será gravada usando o gravador, através de um software de gravação (Requisito Não Funcional – RNF);
- f) o hardware será construído utilizando a linguagem de descrição de hardware Verilog no ambiente de programação Intel Quartus Prime (RNF).

6 METODOLOGIA

Este trabalho será desenvolvido observando as seguintes etapas:

- a) levantamento bibliográfico: pesquisar trabalhos sobre assuntos relacionados, estudar os FPGAs, focar no estudo das linguagens Verilog e VHDL, aprimorar os estudos em arquiteturas de computadores, como o conjunto de instruções da máquina M++ e da máquina M+++, e estudar trabalhos correlatos;
- b) licitação de requisitos: detalhar e reavaliar os requisitos e, se necessário, especificar outros a partir das necessidades observadas durante a revisão bibliográfica;
- c) implementação: implementar o microprocessador proposto, utilizando um kit de desenvolvimento da Altera, que possuindo um chip da família Cyclone IV, utilizando a linguagem de descrição de hardware Verilog e a IDE Intel Quartus Prime. A arquitetura será baseada na Máquina M++ (Borges, 2003) e no projeto de conclusão de curso do André Leonardo Bieging (2018);
- d) os testes: serão baseados em elaborar programas de testes para verificar se o microprocessador está se comportando da maneira esperada, conforme o prometido na ementa do trabalho, processando as entradas e produzindo as saídas corretas no tempo esperado;
- e) otimização: conforme os resultados dos testes, verificar a possibilidade de melhorias na implementação da arquitetura pelo próprio autor. Também é importante deixar documentadas tais possíveis mudanças para que outros autores, em atualizações futuras, possam vir a realizar;
- d) extensão: verificar a viabilidade de adicionar novas instruções, novas operações e melhorias gerais à arquitetura, conforme os resultados dos testes e das otimizações.

7 REVISÃO BIBLIOGRÁFICA

A seguir serão apresentadas as revisões bibliográficas do projeto proposto.

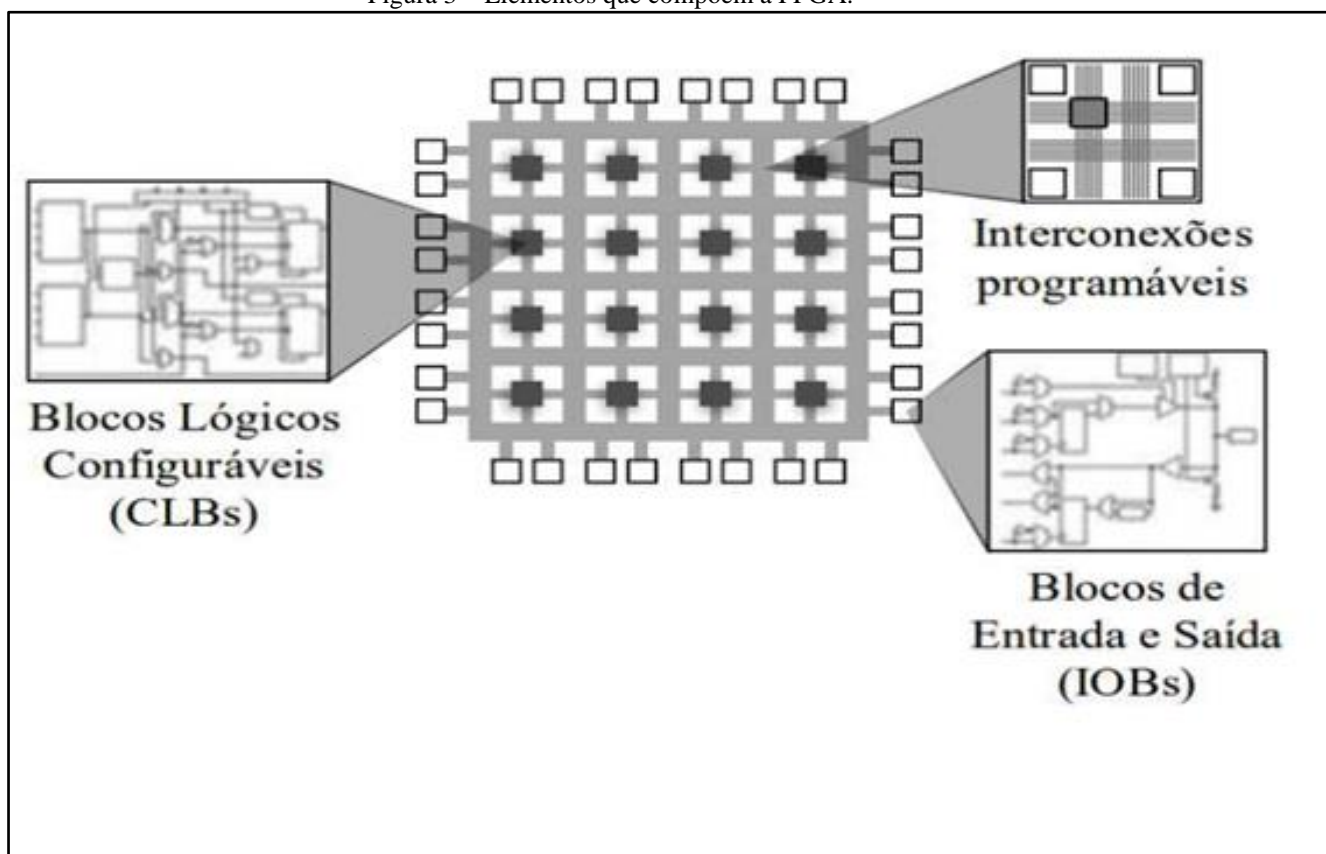
7.1 FPGA

O FPGA consiste em um grande arranjo de células lógicas ou blocos lógicos configuráveis, contidos em um único circuito integrado. Cada célula contém capacidade computacional para implementar lógicas e realizar roteamento para comunicação entre elas. O primeiro FPGA disponível comercialmente foi desenvolvido em 1983 (Costa, 2014).

Barbosa (2019) relata que o FPGA não dispõe de unidade lógica aritmética, assim todas as operações envolvem combinações de blocos conectados eletricamente, que compõem um circuito de hardware customizado. Blocos lógicos podem ser criados, replicados e operar de forma paralela aos demais.

Portanto, conforme Luiz (2021) FPGA é um dispositivo lógico programável constituído por um conjunto de elementos lógicos, dispostos na forma de uma matriz bidimensional, conectados uns aos outros, como apresentado na Figura 3. Além dos elementos lógicos, a FPGA é também composta por outros componentes, tais como blocos de memória dedicada, blocos de entrada e saída, entre outros.

Figura 3 – Elementos que compõem a FPGA.



Fonte: Luiz (2021, p. 30).

Basicamente, a estrutura interna simplificada em um FPGA, segundo Costa (2014), é constituída de blocos lógicos, blocos de entrada e saída, e chaves de interconexão. Os blocos lógicos formam uma matriz bidimensional, e as chaves de interconexão são organizadas como canais de roteamento horizontal e vertical entre as linhas e colunas de blocos lógicos. Esses canais de roteamento possuem chaves de interligação programáveis que permitem conectar os blocos lógicos de maneira conveniente, em função das necessidades de cada projeto.

No entendimento de Taschetto (2020), nos últimos anos, a tecnologia FPGA vem mostrando grandes melhorias, desde o processo de fabricação quanto na melhoria das ferramentas de desenvolvimento disponíveis. Por sua poderosa capacidade de cálculo e flexibilidade, o FPGA é considerado uma solução apropriada para aumentar a capacidade de processamento do inversor.

Assim, dentre as principais características que justificam a utilização desta tecnologia, no entendimento de Luiz (2021) destacam-se: a) a sua alta reconfigurabilidade, a qual torna possível a implementação de todo tipo de circuito digital; b) alta densidade de elementos lógicos, o que permite implementações mais complexas; paralelismo, isto é, circuitos lógicos independentes executados simultaneamente, até mesmo em domínios de clock diferentes; c) alta velocidade de operação, com clock da ordem de Giga-hertz; capacidade de *time to market*, ou seja, o tempo entre o desenvolvimento e a finalização de um produto comercial baseado em FPGA é reduzido, devido a sua capacidade de rápida implementação e execução de testes, e por fim, sua facilidade de sintetização de hardware, através da utilização de linguagens de alto nível, para descrição de hardware.

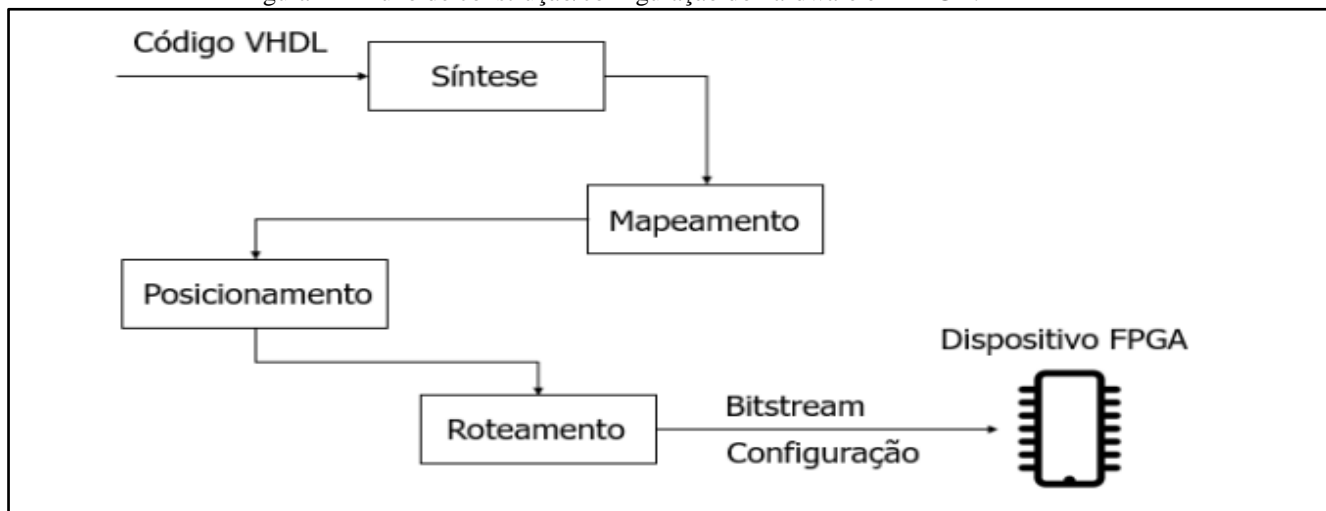
Neste contexto de vantagens dos FPGA, Sousa (2019) comenta que em relação aos ASICs, seria que os primeiros sempre podem ser reconfigurados para formar diferentes circuitos, enquanto que os ASICs não podem ser alterados após serem fabricados, sendo que este processo de fabricação pode exigir grandes investimento financeiros.

Como principais desvantagens, quando comparadas a outros processadores, têm-se o custo, que apesar da popularização dos últimos anos, ainda é maior, e a complexidade de implementação, uma vez que não possui ferramentas de debug, como é o caso de processadores em geral, relata Luiz (2021). Sousa (2019) acrescenta que em contrapartida a flexibilidade do FPGA, implica maior demanda em área de chip, maior *delay* e maior consumo de energia, em relação aos ASICs. Estas desvantagens estão relacionadas à malha de roteamento reprogramável dos FPGAs, sendo que, de modo geral, os FPGAs demandam aproximadamente 20 a 35 vezes mais área que um ASIC padrão, com velocidade 4 vezes menor e consumo de energia 10 vezes maior.

Nesses debates, Taschetto (2020) explica que, desta forma, uma das grandes vantagens do FPGA é a possibilidade do processamento paralelo o qual permite a criação de blocos independentes entre si, consequentemente não exigindo o compartilhamento de recursos e aumentando assim sua capacidade de processamento. Nagai et al. (2023) assinalam que um FPGA é um dispositivo lógico programável, um tipo de circuito integrado que pode ser usado para implementar qualquer circuito digital. Estas características tornam este dispositivo uma excelente ferramenta de estudo e de desenvolvimento para diversas áreas da eletrônica e da computação.

Logo, para a construção do hardware em FPGA, segundo Penha Neto (2021), utiliza-se um software responsável pelas etapas de configuração do dispositivo, conforme descrito na Figura 4.

Figura 4 – Fluxo de construção/configuração do hardware em FPGA.



Fonte: Penha Neto (2021, p. 54).

Conforme a Figura 4 e no entendimento de Penha Neto (2021), a primeira etapa é a sintetização do código fonte em VHDL, em que a sintetização transforma a lógica em alto nível em portas lógicas interconectadas. A próxima etapa é o processo de mapeamento, em que as portas lógicas são separadas em grupos, para serem melhor adaptadas aos recursos lógicos do dispositivo FPGA. Esta etapa é decisiva, pois nela é verificado se é possível a configuração no dispositivo, ou seja, se há blocos lógicos suficientes para construção do hardware. A terceira etapa é o posicionamento, em que ocorre a avaliação sobre qual bloco lógico cada grupo de portas lógicas deve ser configurado. Por fim, a última etapa é o roteamento, em que são definidas as interconexões que irão transportar os sinais entre os blocos. Ao final destas etapas, é gerado um bit stream que contém as informações necessárias para configurar o dispositivo FPGA com a aplicação projetada.

Oliveira (2020) explica, portanto, que as tarefas de processamento do FPGA são enviadas para uma seção dedicada do chip que funciona de modo autônomo sem interferência de outros blocos lógicos. Assim, o desempenho da aplicação não é influenciado com a adição de mais tarefas. O funcionamento da FPGA depende exclusivamente da sua arquitetura. Ela pode operar em ciclos de clock, onde cada operação matemática é executada a cada ciclo. O processamento é realizado em paralelo.

Há disponibilidade de diversos modos para implementações de funções lógicas e matemáticas em cada bloco lógico da FPGA. Em geral, o mais utilizado pelos fabricantes é o LUT, sendo responsável pelas células de armazenamento de memória volátil RAM (*Random Access Memory*), na qual são implementadas de forma otimizada as funções lógicas. Além disso, a linguagem de descrição do hardware (HDL) é popularmente a mais utilizada para programação de FPGA. Entre as principais linguagens HDL, pode-se citar: VHDL (*Very-High Integrated Circuit* HDL) e Verilog, ressalta Oliveira (2020).

Assim, as HDLs são utilizadas no projeto de circuitos eletrônicos, substituindo os diagramas esquemáticos por descrições textuais destes mesmos circuitos. O VHDL foi desenvolvido na década de 1980, para documentar o projeto de circuitos integrados e substituir a utilização de diagramas esquemáticos. O Verilog, também da década de oitenta, foi desenvolvido como uma linguagem proprietária para o projeto de circuitos integrados, descrevem Nagai et al. (2023).

Costa (2014) aponta que as linguagens de descrição de hardware como VHDL foram desenvolvidas para auxiliar os projetistas a documentarem projetos mais complexos, nos quais as descrições por meio de portas lógicas tornaram-se inviáveis. Essas linguagens descrevem o comportamento de circuitos digitais de diversas formas, como funcional, temporal, equações lógicas, tabela verdade, diagramas de forma de onda etc. Barbosa (2019)

comenta que a descrição de hardware do modulador em Verilog pode facilmente ser alterada para operar em outras estruturas. Também pode ser observada a baixa utilização de recursos lógicos do FPGA.

Penha Neto (2021) assinala, portanto, que de certo modo, essas linguagens se aproximam mais de linguagens de desenvolvimento de software e, por isso, costumam facilitar o desenvolvimento de aplicações em FPGAs.

REFERÊNCIAS

- ALMEIDA, Ariane Alves. Construção de um hardware criptográfico do algoritmo DES com a linguagem VHDL em um circuito FPGA. 2012. 83 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação), Departamento de Ciência da Computação, Universidade Federal de Goiás, Catalão, 2012. Disponível em: <https://files.cercomp.ufg.br/webby/up/498/o/Ariane2012.pdf>. Acesso em: 01 nov. 2023.
- BARBOSA, Lucian M. Filtro ativo empregando a célula de comutação de 4 estados para aplicação naval utilizando DSP e FPGA. 2019. 110 f. Dissertação (Mestrado em Engenharia Elétrica), Programa de Pós-Graduação em Engenharia Elétrica, Centro de Ciências Tecnológicas, Universidade Regional de Blumenau, Blumenau, 2019. Disponível em: http://www.bc.furb.br/docs/DS/2019/366604_1_1.pdf. Acesso em: 25 nov. 2023.
- BIEGING, André Leonardo. Implementação da M++ em FPGA. 2018. 74 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação), Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2018.
- BORGES, Jonathan M. Construção de uma UCP hipotética M++. Blumenau, 2003. Disponível em: <http://www.inf.furb.br/~maw/mmaismais/artigos/artigo.pdf>. Acesso em: 01 nov. 2023.
- COSTA, Cesar da. Projetos de circuitos digitais com FPGA. 3 ed. São Paulo: Erica, 2014. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/books/9788536520117>. Acesso em: 25 nov. 2023.
- CUTRESS, Ian. NVIDIA's DGX-2: Sixteen tesla V100s, 30 TB of NVMe, only \$400K. [S.l.], 2018. Disponível em: <https://www.anandtech.com/show/12587/nvidias-dgx2-sixteen-v100-gpus-30-tb-of-nvme-only-400k>. Acesso em: 01 nov. 2023.
- DUTRA, Nayara G. Estudo e implementação em FPGA de um microprocessador. 2016. 77 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica), Departamento de Engenharia Elétrica, Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, 2016. Disponível em: https://www2.dee.cefetmg.br/wp-content/uploads/sites/18/2017/11/TCC_2016_2_NGDutra.pdf. Acesso em: 01 nov. 2023.
- FLACH, Guilherme et al. Revisiting Atari 2600 on an FPGA. In: Viii Southern Conference on Programmable Logic, Bento Gonçalves, Proceedings Piscataway: IEEE, 2012.
- JURGINA, Laura Q. et al. Uno Raid: prototipação em FPGA de um vídeo game desenvolvido em linguagem VHDL. Revista ComIng – Communications and Innovations Gazzete, Minas Gerais, v. 4, n. 1, p. 51-58, 2019. Disponível em: <https://periodicos.ufsm.br/coming/article/view/37747>. Acesso em: 01 nov. 2023.
- LUIZ, Max M. Desenvolvimento de um coprocessador de qualidade de energia padrão Classe A, baseado em processadores embarcados em FPGA. 2021. 94 f. Dissertação (Mestrado em Engenharia Elétrica), Programa de Pós-Graduação em Engenharia Elétrica, Faculdade de Engenharia Elétrica, Universidade Federal de Juiz de Fora, Juiz de Fora, 2021. Disponível em: <https://repositorio.ufjf.br/jspui/handle/ufjf/12472>. Acesso em: 25 nov. 2023.
- MALVINO, Albert P. et al. Digital computer electronics. São Paulo: Career Education, 1992.
- MONTEIRO, Mário A. Introdução à organização de computadores. Rio de Janeiro: Livros Técnicos e Científicos Editora S.A., 1996.
- MOORE, Andrew; WILSON, Ron. FPGAs for dummies, 2 ed. Wiley: 2017.
- MONTFORT, Nick; BOGOST, Ian. Racing the beam: the Atari Video Computer System. Cambridge, London: The MIT Press, 2009
- NAGAI, Diego S. et al. Portabilidade e customização de um Softcore Risc-V em FPGA. Revista Brasileira de Mecatrônica, São Caetano do Sul, v. 5, n. 4, p. 77-105, abri./jun. 2023. Disponível em: <https://revistabrmecatronica.sp.senai.br/ojs/index.php/revistabrmecatronica/article/view/212/163>. Acesso em: 25 nov. 2023.
- OLIVEIRA, Jonathan de. Implementação de um sistema de transferência de dados de ultrassom via etherneti e processamento embarcado em dispositivo FPGA. 2020. 109 f. Dissertação (Mestrado em Engenharia Elétrica), Programa de Pós-Graduação em Sistema de Energia, Universidade Tecnológica Federal do Paraná, Curitiba, 2020. Disponível em: <http://repositorio.utfpr.edu.br/jspui/bitstream/1/5100/1/ultrassomethernetembarcadofpga.pdf>. Acesso em: 25 nov. 2023.
- PENHA NETO, Gérson. Navegação autônoma de vant por fusão de dados com rede neural artificial otimizada implementada em FPGA. 2021. 215 f. Tese (Doutorado em Computação Aplicada), Curso de Pós-Graduação em Computação Aplicada, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2021. Disponível em: <http://mtc-m21c.sid.inpe.br/col/sid.inpe.br/mtc-m21c/2021/01.28.22.11/doc/publicacao.pdf>. Acesso em: 25 nov. 2023.

SOUSA, Mark C. F. Módulo para execução de redes neurais convolucionais em FPGA. 2019. 118 f. Dissertação (Mestrado em Ciências), Escola Politécnica, Departamento de Engenharia de Sistemas Eletrônicos, Universidade de São Paulo, São Paulo, 2019. Disponível em: <https://teses.usp.br/teses/disponiveis/3/3140/tde-14082019-110912/publico/MarkCappelloFerreiradeSousaCorr19.pdf>. Acesso em: 25 nov. 2023.

TASCHETTO, Rodrigo L. Análise e desenvolvimento de modulação vetorial espacial em FPGA aplicado em inversor trifásico multinível NPC. 2020. 157 f. Dissertação (Mestrado em Engenharia Elétrica), Programa de Pós-Graduação em Sistemas de Energia, Universidade Tecnológica Federal do Paraná, Curitiba, 2020. Disponível em: <http://repositorio.utfpr.edu.br/jspui/bitstream/1/5281/1/modulacaovetorialinversortrifasico.pdf>. Acesso em: 25 nov. 2023.

VAUGHAN-NICHOLS, Steven. A super-fast history of supercomputers: from the CDC 6600 to the sunway taihuLight. [S.l.], 2017. Disponível em: <https://www.hpe.com/us/en/insights/articles/a-super-fast-history-of-supercomputers-from-the-cdc-6600-to-the-sunway-taihuLight-1711.html#>>. Acesso em: 01 nov. 2023.

VIANA, Eveli M. et al. Valorização acadêmica no processo de formação de um verdadeiro cientista da computação. In: WEAC 2009, p 103-106, out. 2009.