



PLATAFORMA DESENVOLVIDA EM SIMULADOR PARA FACILITAR A APRENDIZAGEM DA DISCIPLINA DE ARQUITETURA AVANÇADA PARA COMPUTAÇÃO

Jordan da Silva Floriano - jordan.floriano@cear.ufpb.br
Universidade Federal da Paraíba, Departamento de Engenharia Elétrica
Cidade Universitária - SN
CEP: 58051-900 - João Pessoa - PB - Brasil
José Maurício R. de Souza Neto - mauricio@cear.ufpb.br

***Resumo:** A disciplina de arquitetura avançada para computação, na Universidade Federal da Paraíba, é dividida em três estágios. Nos primeiro e no segundo estágio o professor da disciplina utiliza como método de avaliação um projeto de um processador de ciclo único. Esse processador deve obedecer algumas especificações dadas pelo professor e precisa ser feito desde a concepção do hardware até o software. Porém, o professor da disciplina vem percebendo que os alunos não conseguem, em sua maioria, entregar a atividade no prazo estimado, e isso se deve ao fato de que muitos passam uma boa parte do tempo na fase inicial do projeto. Fase essa que exige um bom conhecimento de assunto de disciplinas anteriores. Para tentar diminuir esse problema, elaborou-se com o auxílio do Proteus® um projeto de processador idêntico ao que se deve ser implementado na disciplina, para que os alunos pudessem aproveitar a estrutura básica já montada e assim poder aplicar os conhecimentos da disciplina de forma mais direta, diminuindo o tempo na fase inicial do projeto. Com esse projeto espera-se que haja um aumento na produtividade de alunos dos próximos semestres dessa disciplina.*

***Palavras-chave:** Arquitetura avançada, Processadores, Proteus.*

1 INTRODUÇÃO

A disciplina arquitetura avançada para computação (AAC), da Universidade Federal da Paraíba (UFPB), faz parte da grade curricular do curso de engenharia elétrica, sendo uma disciplina obrigatória na ênfase de Eletrônica. Esta disciplina tem um caráter peculiar, devido ao fato de abordar conceitos de várias outras disciplinas do curso. Isso faz com que o aluno tenha que ter um bom conhecimento das disciplinas anteriores para que não sinta tanta dificuldade.

Uma das formas de avaliações dessa cadeira é fazer com que os alunos implementem um processador, o qual é melhorado durante os três estágios da disciplina. Porém, foi detectado pelo professor que os alunos muitas vezes não conseguiam entregar o trabalho a tempo, por dificuldades na execução. Muitas vezes, essas dificuldades estavam mais ligadas aos conhecimentos prévios necessários de que aos conhecimentos novos, aprendidos na disciplina. Neste trabalho descreve-se uma plataforma ilustrativa, concebida a partir de projeto de um dos alunos, que possibilita a implementação dos conhecimentos da disciplina



de forma mais rápida, já que o aluno não se preocupará em fazer o processador desde o início. Essa plataforma foi projetada utilizando o programa ISIS®, do pacote Proteus®.

2 SOBRE A DISCIPLINA

O foco principal da disciplina AAC é o estudo sobre os processadores, desde sua arquitetura física até a sua programação, ou seja, do *hardware* ao *software*. Um processador é um dispositivo capaz de realizar, a partir de um conjunto de instruções, cálculos lógicos e aritméticos de alta complexidade (VON NEUMANN, 1945). É uma das partes desse computador é o processador. Existem diversas arquiteturas e abordagens sobre processadores, a ementa da disciplina é baseada na arquitetura RISC.

A disciplina pode ser dividida em três estágios: processadores de ciclo único, processamento paralelo no nível de *software* e no nível de *hardware*. Na parte de processadores de ciclo único são apresentados aos alunos os conceitos de escrita e leitura de registradores, acesso a memória, instruções e linguagem de máquina. Esses processadores têm como principal característica que todas as suas instruções levam apenas um ciclo para execução. Essa é a primeira atividade passada para os alunos: desenvolver um processador de ciclo único que seja capaz de executar um determinado conjunto de operações.

Após os conceitos iniciais de processadores é introduzida a idéia de paralelismo em nível de *hardware*. Isso significa que o processador consegue realizar mais de uma instrução ao mesmo tempo, devido a alterações feitas no *hardware*. Uma forma de realizar essa alteração é implementando a técnica de *pipeline*. Esta técnica consiste em dividir o sistema em estágios, e entre esses estágios a informação é retida por registradores, fazendo com que a informação possa ser mantida inalterada ou possa ser passada para estágios subsequentes, dependendo da instrução (QUINN, 2004). A segunda parte da atividade consiste em implementar o *pipeline* no processador desenvolvido no estágio anterior.

O último estágio da disciplina aborda o tratamento via *software* das questões de paralelismo. Neste estágio é visto como o sistema operacional pode gerenciar as tarefas de forma que o uso do processador seja aprimorado. Esse aprimoramento é alcançado através de programações com *threads* e gerenciadores, que controlam o acesso múltiplo a memórias e a carga de trabalho de cada núcleo do processador. A avaliação desses tópicos não é feita mais utilizando o processador ilustrativo dos outros estágios e sim utilizando programas e simuladores específicos.

3 O PROCESSADOR

Nesta seção serão apresentados os processadores desenvolvidos nos dois primeiros estágios da disciplina. Será apresentado o *software* utilizado e suas principais ferramentas que auxiliaram no projeto. Serão mostradas suas principais características que devem ser implementadas e as partes que os compõe.

3.1 Simulador ISIS®

Proteus® Design Suite é um *software* de simulação e concepções de circuito, desenvolvido pela empresa Labcenter Electronics®. O Proteus® inclui uma ferramenta de simulação de circuitos eletrônicos, chamada de ISIS®. Essa ferramenta permite não só a simulação de componentes passivos e ativos, mas também a simulação de circuito integrado programáveis, microcontroladores.

Organização



Promoção

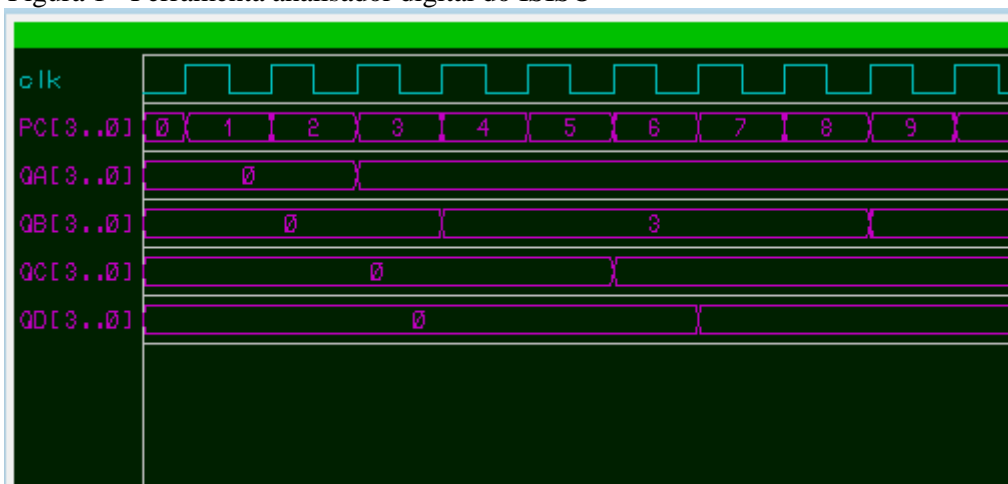




O usuário do ISIS® tem acesso a diversos componentes ideais e reais. Os componentes reais simulam as características dos componentes comercializados. Com isso, é possível obter resultados que servem como base para uma montagem real. O simulador também possui ferramentas que emulam os instrumentos de medidas, como voltímetros, osciloscópios analisador de frequências, entre outros.

Além de ser prático e simples de se usar, um dos motivos para a escolha do ISIS® para a elaboração desse projeto é que ele tem ferramentas que são de grande valia na avaliação do funcionamento de um processador, como o analisador digital, que fornece um gráfico com as formas de ondas digitais de pontos específicos do circuito. O gráfico gerado por essa ferramenta (Figura 1) permite ao usuário acompanhar o funcionamento detalhado do circuito, sendo possível determinar em que ponto ocorreu um erro, por exemplo.

Figura 1 - Ferramenta analisador digital do ISIS®



Outra ferramenta que torna interessante o uso do ISIS® para o desenvolvimento desse projeto é que ele pode fazer simulações em tempo real. Para a aplicação de um processador, significa que será possível acompanhar as mudanças nos diferentes blocos do projeto. Esse acompanhamento pode ser realizado com a ajuda de um *display* que converte de código binário para decimal, fazendo com que a visualização fique mais simples e objetiva.

3.2 Processador sem pipeline

Como dito anteriormente, a primeira avaliação da disciplina consiste na elaboração de um processador com arquitetura RISC. Esse processador tem como característica executar uma instrução em um único ciclo. Para realizar essa parte da avaliação o professor da disciplina exige que o processador seja capaz de executar determinadas funções, que podem ser vistas na Tabela 1.

Tabela 1 - Instruções do processador do primeiro estágio

Instrução	Descrição
Move	Copia o conteúdo de um registrador em outro registrador
Soma	Soma o valor de um registrador ao valor de outro registrador
Subtrai	Subtrai o valor de um registrador ao valor de outro registrador
Soma imediato	Soma o valor do dado ao valor de um registrador
Subtrai imediato	Subtrai o valor do dado ao valor de um registrador
Multiplicação	Multiplica o valor de um registrador por 2

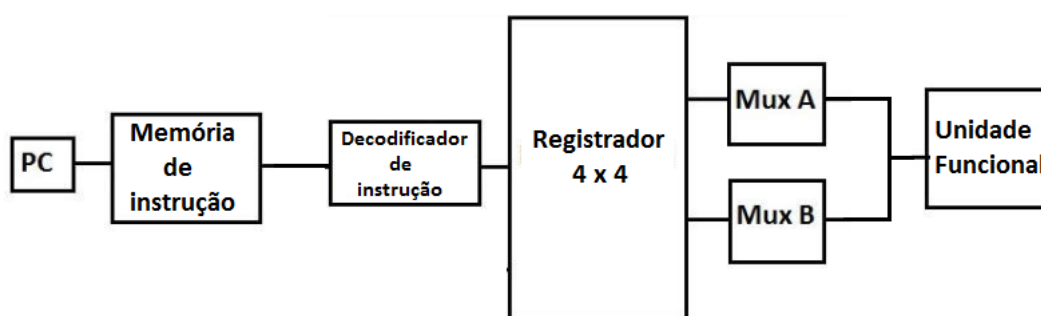


Divisão	Divide o valor de um registrador por 2
Limpa	Zera o conteúdo de um registrador
Pulo	Altera a sequência de execução das instruções

A partir dessas instruções, cabe a cada aluno desenvolver uma lógica de organização para implementar nos seus processadores. Essa lógica é pensada de acordo com o *hardware* que será implementado. Após essa parte, o aluno terá um conjunto de instruções e seus respectivos códigos de máquina.

Para o conjunto de instruções dados o processador foi estruturado de acordo com a Figura 2, onde se podem observar os principais blocos do processador, como a unidade funcional e o decodificador de instrução.

Figura 2 - Diagrama em bloco do processador do primeiro estágio



A unidade funcional engloba a unidade lógica aritmética (ULA), que é responsável pelos cálculos lógicos e aritméticos do processador. A parte aritmética é constituída por um circuito somador com um circuito lógico combinacional em uma de suas entradas, para que a partir dessa lógica ele realize outras operações além da adição (MORRIS *et al.*, 2015). Para a elaboração desse circuito é necessário que o aluno possua um conhecimento prévio sobre somadores e mapas de Karnaugh, porque será preciso levantar uma tabela verdade e a partir dela, após simplificações, gerar o circuito necessário.

O registrador 4x4 é outro bloco importante no processador, nele está contido os registradores que armazenam os operandos e os resultados das operações da ULA. Para isso é necessário um circuito demultiplexador que controle em qual registrador determinada informação será salva.

No contador de programas, assim como na ULA e no bloco funcional, também é necessário a utilização de circuitos lógicos para que o circuito realize a função esperada. Com esses exemplos fica clara a dependência dessa disciplina a assuntos ensinados nas disciplinas anteriores de circuitos lógicos e arquitetura de computadores.

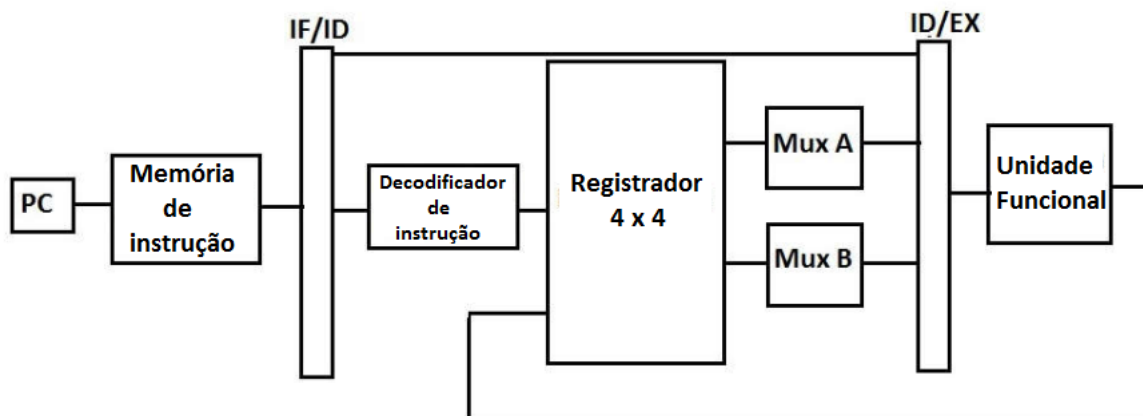
3.3 Processador com *pipeline*

O processador explicado nessa seção é o que foi pedido pelo professor como avaliação do segundo estágio da disciplina, ou seja, o processador com *Pipeline*. Na Figura 3 pode-se observar o diagrama em blocos do processador com *pipeline*, que se diferencia do diagrama do primeiro estágio apenas pela inserção dos registradores de *pipeline*. O *pipeline* utilizado nesse processador foi um *pipeline* de três estágios, cada estágio com sua função específica:



- **Estágio de busca de instrução (*Instruction fetch* – IF)** – Acessa a memória de instrução no endereço indicado pelo contador de programa (*program counter* – PC) e busca a instrução a ser executada.
- **Decodificação de instrução (*Instruction Decoder* – ID)** – Decodifica a instrução e ler o conteúdo de um dos registradores do bloco registrador de arquivos. A decodificação é feita de forma paralela a leitura dos registradores, o que é possível porque os especificadores dos registradores estão em locais fixos nas instruções de uma arquitetura RISC. (HENNESSY & PATTERSON, 2012)
- **Execução e escrita (*Execution + Write-back* – EX+WB)** – Executa a operação na ULA com os operandos carregados no estágio anterior. Dependendo do tipo de instrução, a operação pode acontecer entre registradores, com registradores e imediatos ou com acesso direto a memória. Após a execução da operação o resultado é escrito no registrador de destino.

Figura 3 - Diagrama em bloco de um processador com *Pipeline*



Toda instrução executada pelo processador passará por esses três estágios, porém, em alguns casos, a próxima instrução pode precisar de uma informação que ainda não foi computada. Para corrigir esse problema é necessário o uso de uma bolha, um tempo em que o processador gasta um ciclo de *clock* sem realizar nenhuma operação. Essa bolha serve para dar tempo de uma informação ser gravada nos registradores de *pipeline* IF/ID e ID/EX.

4 VANTAGEM DA PLATAFORMA

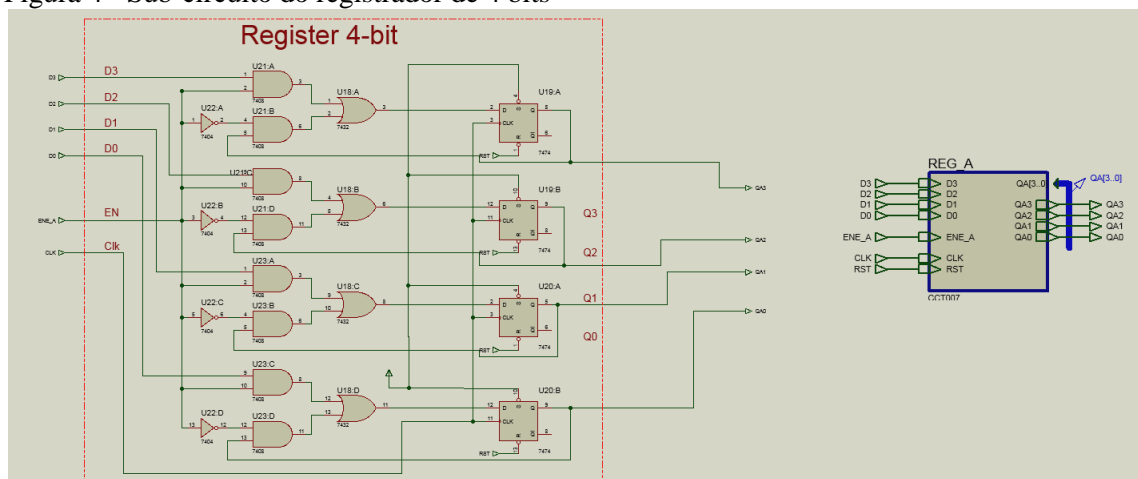
A plataforma proposta nesse trabalho serve para trazer facilidades durante o processo do desenvolvimento das atividades da disciplina. O fato de ela ser construída com blocos seguindo a metodologia do livro base da disciplina e de que cada bloco desses pode ser atualizado e testado separadamente.

Na hora do desenvolvimento do processador percebeu-se que seria necessário repetidamente o uso de registradores e multiplexadores que, devido ao número específico de canais de entrada e saída, não constavam na biblioteca do ISIS®. Portanto, um dos primeiros passos tomados com o pensamento de facilitar o uso futuro desse ambiente foi construir um sub-circuito, como é chamado no ISIS®, para os registrados e multiplexadores. Isso torna a



utilização e replicação desses circuitos mais fáceis. O sub-circuito do registrador de 4 bits pode ser visto na Figura 4.

Figura 4 - Sub-circuito do registrador de 4 bits



Nos sub-circuitos dos registradores e dos multiplexadores, usaram-se apenas conhecimentos de circuitos lógicos combinacionais para fazer as funções pretendidas. E é com essa intenção que essa plataforma é disponibilizada. O aluno de AAC que estiver projetando esse processador não precisa mais criar esses blocos, o que demanda tempo e, devida a quantidade de ligações, aumenta a chance de erros de montagem, causando assim um desperdício de tempo que poderia está sendo utilizado para o desenvolvimento, de fato, do processador com as características requeridas pelo professor.

Os dois sub-circuitos citados anteriormente referem-se a conteúdos que já foram aprendidos anteriormente por isso seria útil substituírem por blocos. Mas nessa plataforma a ferramenta de sub-circuito do ISIS® também foi explorada com a intenção de deixar a visualização e a configuração do processador mais simples e modular. Os sub-circuitos foram utilizados para criar módulos do processador, como a unidade funcional, a unidade lógica e os registradores de arquivos.

A partir de essa estrutura modular e seguindo como base o livro referencia da disciplina, o aluno que está projetando o processador pode ter acesso direto à determinada parte do processador para fazer melhorias e mudanças que se adéquem melhor a seu respectivo projeto. Por exemplo, se o estudante quiser adicionar mais um registrador ao seu projeto de processador ele pode acessar o bloco do registrador de arquivo e adicionar mais um sub-circuito de registrador, apenas fazendo as ligações necessárias.

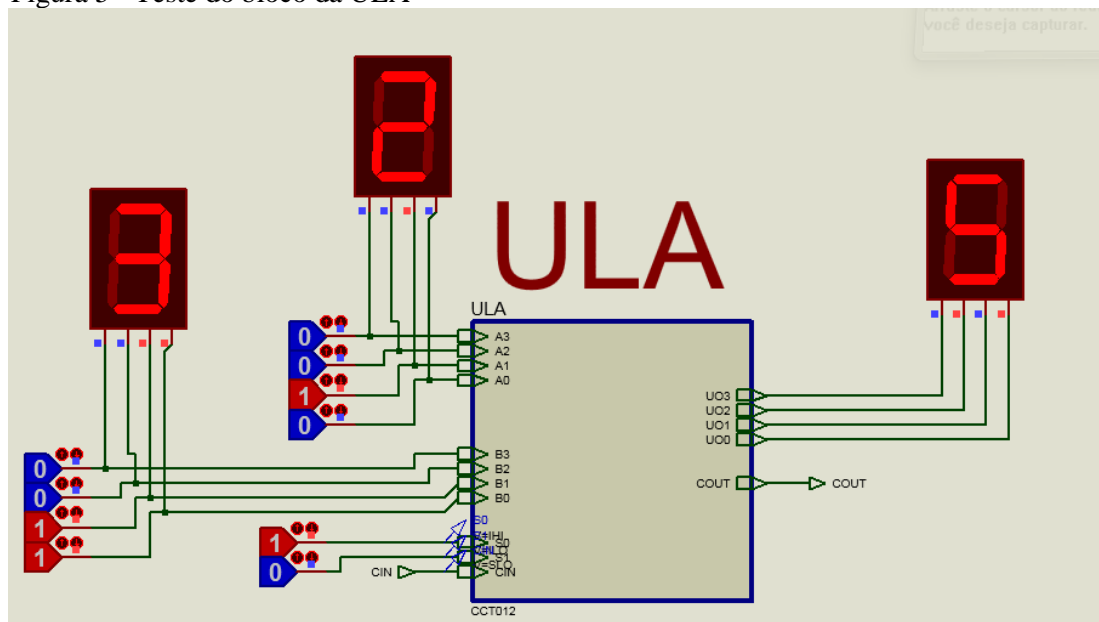
Uma outra vantagem proporcionada por essa plataforma é facilidade de realizar testes nos módulos do processador. Dessa forma, o aluno pode testar individualmente o bloco toda vez que ele fizer uma alteração e conferir se ele está funcionando como esperado. Isso facilita na hora de montar todos os blocos do processador, porque o aluno tem a garantia que cada bloco está funcionando bem individualmente. Caso o processador não funcione como esperado o aluno pode concluir que o problema pode ser na interconexão dos blocos ou na teoria aplicada.

Como exemplo, pode-se analisar um caso em que é preciso saber se a operação de soma está sendo executada corretamente pelo processador. Para isso o aluno pode acessar o bloco que é diretamente responsável por essa operação e fazer o teste, nesse caso a ULA. Após acessar o bloco é possível colocar indicadores de estados digitais, uma ferramenta de simulação do ISIS®, que iram fornecer a entrada ao sistema. Esses indicadores também



podem ser colocados nas entradas de configurações e seleção do bloco. O resultado pode ser visto de forma rápida, porque já é feita a conversão de binário pra decimal, em um *display* sete segmentos. Na Figura 5 pode-se observar um teste realizado na ULA, onde são colocados os números 2 e 3 na entrada e é selecionada a operação de soma.

Figura 5 - Teste do bloco da ULA



5 CONSIDERAÇÕES FINAIS

Foram apresentados alguns detalhes e características da implementação de um processador RISC no ambiente de simulação ISIS®, do pacote de *software* Proteus®. Essa implementação mostrou-se confiável e de boa usabilidade para a aplicação dos conceitos da disciplina de arquitetura avançada para computação. Provando, devida a sua objetividade, que facilita a resolução dos problemas de projeto pedido pelo o professor.

Percebe-se que a aplicação desse projeto na disciplina de AAC pode fazer com que a média de alunos que conseguem finalizar os trabalhos da disciplina no prazo aumente, já que eles poderão utilizar os blocos e montar-los já da forma que melhor lhe convir. Com essa fase inicial desenvolvida mais rapidamente abre espaço para que o professor consiga inserir novas atividades que possam andar em paralelo com a teoria abordada em sala.

Para trabalhos futuros pretende-se fazer melhorias no manuseio dessa plataforma. Como por exemplo, inserir esses sub-circuitos e módulos na biblioteca do ISIS® para que eles possam ser trabalhados como qualquer outro componente. Além de preparar um material com as instruções básicas para a utilização dos sub-circuitos e das características técnicas de cada bloco.

REFERÊNCIAS BIBLIOGRÁFICAS

HENNESSY, J. L. & PATTERSON, D. A. Computer architecture: a quantitative approach. 2ed Morgan Kaufmann Publishers, Inc. San Francisco, 998p. 1996.

MORRIS, Mano; KIME, Charles; MARTIN, Tom. Logic and computer Design Fundamentals. 5ed. Pearson Prentice Hall, 666p. 2015

Organização



Promoção





QUINN, M. Parallel Programming in C with MPI and OpenMP. 2ed. McGraw-Hill, 120p. 2003.

VON NEUMANN, J. **First draft of a report on the EDVAC** Disponível em:
<<http://www.virtualtravelog.net/wp/wp-content/media/2003-08-TheFirstDraft.pdf>> Acesso em: 24 mai. 2017.

PLATFORM DEVELOPED IN SIMULATOR TO FACILITATE THE LEARNING OF THE DISCIPLINE OF ADVANCED ARCHITECTURE FOR COMPUTING

Abstract: *The discipline of advanced architecture for computing, at the Federal University of Paraíba, is divided into three stages. In the first and second stage the teacher of the discipline uses as a method of evaluation a single-cycle processor design. This processor must meet certain specifications given. However, the teacher of the discipline has realized that the majority of the students are unable to deliver the activity in the estimated time, and this is due to the fact that many spend a good part of the time in the initial phase of the project. This phase requires a good knowledge of the subject of previous disciplines. To try to reduce this problem, a processor project similar to what was implemented in the discipline was elaborated with the aid of Proteus®, so that the students could take advantage of the basic structure already assembled and thus be able to apply the knowledge of the discipline in a more objective way, Reducing the time in the initial phase of the project. With this project it is expected that there will be an increase in the productivity of students of the next semesters of this discipline.*

Key-words: *Advanced architecture, Processors, Proteus.*

Organização



Promoção

