

An improved version of the didactic computer for teaching digital electronics and microprocessor concepts

Matheus Fernando Tasso

Depart. de Eng. Elétrica

Instituto Federal do Paraná - Paranavaí

Paranavaí, PR, Brasil

matheustasso2011@hotmail.com

Carlos Daniel De Souza Nunes

Depart. de Eng. Elétrica

Instituto Federal do Paraná - Paranavaí

Paranavaí, PR, Brasil

carlosnunes9852@gmail.com

Edson Junior Acordi

Depart. de Eng. Elétrica

Instituto Federal do Paraná - Paranavaí

Paranavaí, PR, Brasil

edson.acordi@ifpr.edu.br

Abstract—In teaching microprocessors and microcontrollers, it is necessary to know the fundamentals of digital electronics and low-level languages(assembly). In general, this proves to be a great challenge for both those who teach and those who learn. In teaching, microcontrollers are usually used with all their peripherals already built in, and this usually does not require a very large level of abstraction from the student to be understood. To solve this problem, the following project aims to facilitate and make teaching this subject more fun, building the computer and its peripherals using only discrete components and integrated circuits (flip-flops, registers, logic gates, etc.). The proposed computer hardware was built and simulated in Digital software.

Index Terms—microprocessor, microcontroller, assembly language, educational computer.

I. INTRODUÇÃO

A era dos computadores modernos se deu a partir da criação do ENIAC (Electronic Numerical Integrator and Computer) em 1946 por John Presper Eckert e John William Mauchly na universidade da Pensilvânia. O ENIAC foi o primeiro computador eletrônico digital de uso geral e era constituído de aproximadamente 18000 diodos termiônicos (válvulas eletrônicas ou válvula de elétrons) e mais 1500 reles, pesando 30 toneladas e consumindo uma potência de 140 kW [1], [2]. O ensino do conteúdo sobre microprocessadores e microcontroladores, por muito tempo vem sendo enfrentado tanto por professores quanto por estudantes como sendo um grande desafio devido à natureza do conteúdo, o qual combina o projeto de circuitos digitais com a linguagem de programação assembly [3], [4], [5], [6]. É fato que o estudo detalhado do funcionamento de sistemas microcontrolados e/ou microprocessados na engenharia elétrica é de fundamental importância, podendo ser um fator de grande impacto tanto na carreira profissional como acadêmica. A importância se torna ainda mais evidente, se for considerado que, atualmente a grande maioria das plataformas de ensino no contexto dos microprocessadores, são sistemas constituídos basicamente por microcontroladores, como, por exemplo, o Arduino que é uma plataforma de prototipagem eletrônica de código aberto [7], [8], [4]. Até mesmo o uso de microcontroladores como o PIC

da Microchip, o AVR da Atmel e o ARM da ST são muito utilizados para o ensino dos conceitos de microprocessadores e microcontroladores. Nestes casos, o estudo individual dos elementos essenciais dos processadores/computadores (unidade lógica e aritmética, memória, endereçamento, geração de sinais de controle, contador de programa, registradores, etc.) se torna mais complexo, pois grande parte dos elementos do hardware deve ser abstraído, deixando assim uma lacuna no entendimento sobre a arquitetura e projeto de processadores/computadores, logo, perdendo-se parte da essência do conhecimento sobre estes dispositivos [5], [9]. É importante destacar que, esse problema poderia ser minimizado se plataformas para estudo prático fossem empregadas, como, por exemplo, os antigos microprocessadores como o Z80 da Zilog, o 6502 da antiga MOS/CSG, o 8086 da Intel, ou até mesmo o microcontrolador 8051 da Intel, nos quais vários periféricos externos são necessários para se construir um computador, logo, exigindo um estudo em maior profundidade para se entender o funcionamento do sistema, o que resultaria no nível de conhecimento desejado para os componentes curriculares que estudam microcontroladores, microprocessadores e computadores. Assim, para suprir essa lacuna, no intuito de tornar o processo de ensino/aprendizagem mais eficiente, este projeto se propõe a estudar, desenvolver, simular e implementar um computador simples e de baixo custo para fins educacionais com foco na consolidação das habilidades e conceitos fundamentais sobre microcontroladores, microprocessadores e computadores. Cabe destacar que muitos métodos têm sido propostos para se obter resultados satisfatórios no ensino do conteúdo de microprocessadores e microcontroladores de modo a proporcionar ao estudante um entendimento suficiente. Em [10], a eficácia no uso de simuladores para o ensino de microprocessadores e microcontroladores foi discutida, no qual os autores concluem que simuladores são de fundamental importância para um melhor entendimento deste conteúdo. Já no contexto do uso de hardware dedicado ao ensino de microprocessadores e microcontroladores, existem muitas propostas no meio acadêmico. Um computador simples de 4 bits que faz uso da unidade lógica e aritmética 74LS181 foi

proposto em [11], [12], Em [13] o projeto de um processador simples para fins educacionais foi desenvolvido em VHDL e implementado em um FPGA. Uma versão estendida desse processador foi elaborada em [14], a qual foi denominada de LCP (Low Complexity Processor). É fato que o uso de um FPGA poderia ser adotado para contornar parcialmente as dificuldades mencionadas no ensino do tema abordado, no entanto, as plataformas com FPGA tendem a ter maior custo do que o projeto proposto e além disso, não são de fácil manutenção, como por exemplo, se acidentalmente um I/O for danificado, este é perdido, pois quase tudo está concentrado em um circuito integrado com encapsulamento SMD (surface mount device), de modo que, o reparo, quase sempre se torna inviável. O projeto aqui desenvolvido, é uma melhoria do computador proposto em [16], que teve como base e ponto de partida o projeto de [15], o qual consiste em um pequeno computador de 4 bits, com uma notável qualidade didática.

II. MATERIAIS E MÉTODOS

Para a definição da arquitetura do projeto, optou-se por um computador com um tamanho de palavra de 4 bits, de modo que ficasse o mais simples possível, no entanto, sendo flexível para a criação de programas. Cabe mencionar que, por questões de simplicidade, optou-se por não implementar instruções para chamada/retorno de sub-rotina, a qual permite a possibilidade de reutilizar um mesmo código em várias partes do programa, no entanto, implicando em hardware adicional. A arquitetura proposta para o computador de 4 bits, constitui de um acumulador (ACC), três registradores, quatro portas de saída (OUTA, OUTB, OUTC e OUTD), quatro portas de entrada (INA, INB, INC e IND), memória RAM, um contador de programa de 11 bits (PC – Program Counter), memória de programa de 2k x 16 e decodificador de instruções. Além disso, é possível notar o circuito de clock, a unidade de controle (UC), responsável por gerar todos os sinais que controlam o fluxo de dados e por fim, a unidade lógica e aritmética (ULA). Cabe mencionar que, optou-se pela implementação apenas da operação lógica NAND, no entanto, todas as operações lógicas podem ser executadas por software, considerando que a NAND é universal, logo, sendo possível obter as outras operações. A seguir, o computador é completamente descrito em sete partes por meio de diagramas esquemáticos. A Figura 1 ilustra o diagrama do contador de programa e da memória de programa. O contador de programa é formado por três contadores binários 74161, que associados possuem um grau de relevância. O reset dos contadores é realizado por um sinal que vem do circuito de reset da Figura 4. O sinal para o contador realizar saltos é obtido do circuito montado com um 74HC74 associado a uma combinação de portas lógicas e o endereço do salto é informado pelas entradas: LA0 até LA3 do contador PCL; MA0 até MA3 do contador PCH; e RD0 até RD3 do contador PCH. A memória de programa é formada pela RAM 6116 que tem organização de 2k x 8, logo, foram utilizadas duas em paralelo para permitir o uso de uma palavra de 16 bits, necessários para armazenar o código das instruções. É possível

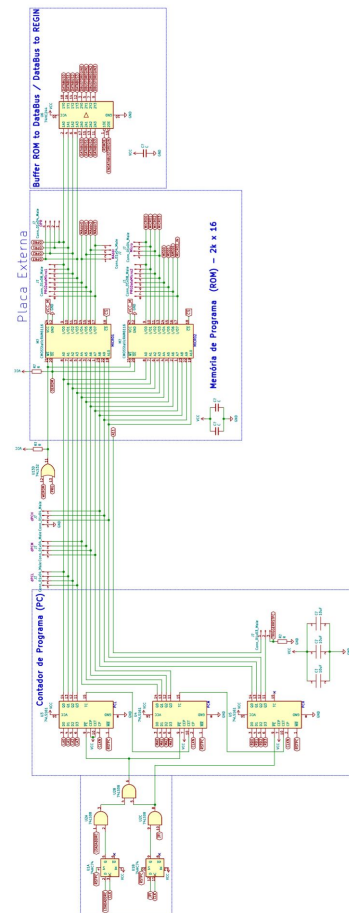


Fig. 1. Contador e Memória de Programa

ainda observar um buffer (74244), que permite a conexão do operando n com o barramento de dados (DATABUS). O mesmo buffer é utilizado para a conexão do DATABUS com os registradores e portas de saída. Cabe ainda mencionar que todos os conectores próximos aos nomes que iniciam com d são utilizados para apenas a visualização do valor em um display, por exemplo, dPCL, dPCM, dRB, etc. O decodificador de instruções ilustrado na Figura 2 é formado por dois 74138, que é um decodificador de 3 para 8 linhas. O sinal MICRO3 executa a seleção de qual decodificador é utilizado e os sinais MICRO0, MICRO1 e MICRO2 são decodificados para a seleção da instrução. Na Figura 3 é ilustrado o circuito de controle do computador, onde os sinais para a execução das instruções são gerados. Os sinais para as instruções LDI, LDR e LDA são gerados com um circuito combinacional que utiliza portas lógicas AND e OR feitas com diodos e os circuitos integrados OR (7432) e AND (7408). Os sinais gerados são LDIACC n , DATATOREG, RALE, RBLE, RDLE, RAOE, RBOE, RDOE. Para as instruções OUTA, OUTB, OUTC e OUTD, a geração dos sinais OUT xn , OUT xRA , OUT $xRAM$, DATATOREG2 e ENDATABUSTOREGIN é realizada com um circuito integrado de portas OR (7432), uma porta NOT (7404) e uma AND com diodos. Para a execução da instrução

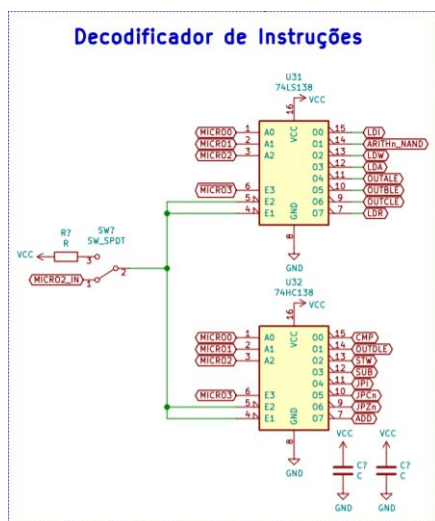


Fig. 2. Decodificador de Instruções

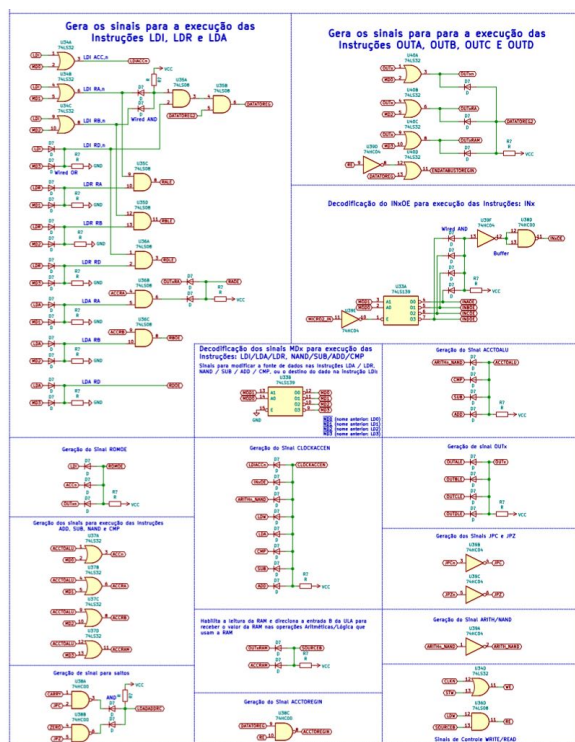


Fig. 3. Unidade de Controle

de entrada, INx, é realizada a decodificação dos sinais MOD0 e MOD1, que habilita uma das quatro portas de entrada. O decodificador 2x4 utilizado é um 74139. O sinal de controle que permite a passagem do valor do acumulador para a ULA é o ACCTOALU, que é gerado por uma porta AND de quatro entradas feita com diodos. Os sinais de entradas são ARITHn_NAND, CMP, SUB e ADD. A geração do sinal ROMOE, CLOCKACCEN e OUTx também são realizadas com portas AND construídas com diodos. Ainda na Figura 3, temos a geração de sinais para a realização de saltos

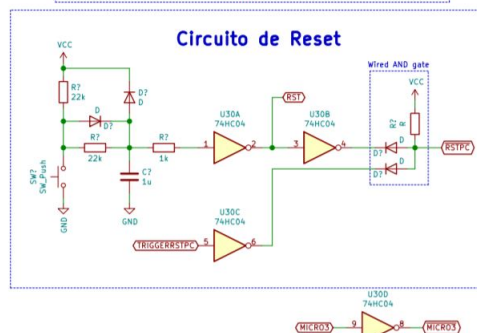
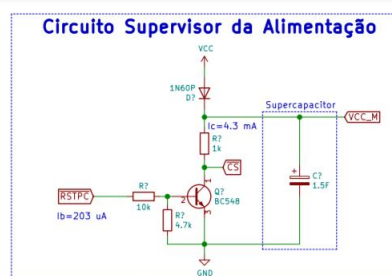


Fig. 4. Circuito Supervisor de Alimentação e Circuito de Reset

no contador de programa, sinais que habilitam as outras instruções e outros sinais de controle. A Figura 4 ilustra uma das modificações em relação ao circuito [16]. Nela temos um circuito supervisor de alimentação, que permite que o código escrito na memória de programa permaneça gravado por um período de tempo maior, possibilitando que o programa gravado não seja perdido no desligamento do computador. Também é demonstrado o Circuito de Reset que gera o sinal RSTPC, responsável por reiniciar o contador de programa do computador. Na Figura 5 o circuito de clock e os osciladores são ilustrados. O circuito de controle de clock foi projetado com nove portas lógicas, quatro portas AND (7408), quatro portas NOT (7404) e uma porta OR (7432). Foi projetado com uma lógica combinacional para entre o clock automático e o clock manual. Para sinal STEP do clock manual, foi utilizado o circuito integrado timer 555 para fazer o debounce. Para a base de tempo do clock automático, foi optado por duas opções diferentes, a base de tempo fixa e precisa, usando um Pierce oscillator feita com um cristal de quartzo, e a base de tempo com a frequência ajustável projetada com o timer 555 na configuração astável. Temos a ULA na Figura 6. Em relação a outra versão, poucas alterações foram realizadas e uma delas é que os bits para entrada na ULA podem ser do barramento de dados ou da memória RAM. As operações realizadas pela ULA são adição, subtração, comparação e a operação lógica NAND, que é uma operação lógica que permite a realização de todas as outras operações lógicas. A seleção da entrada da ULA é realizada pelo 74157, que é um seletor de duas palavras de quatro bits. O complemento de bits para a realização da subtração é realizado pelo 7486 (XOR). Outro MUX (74157) é utilizado para selecionar a origem da operação (lógica ou aritmética). Um somador binário (74283)

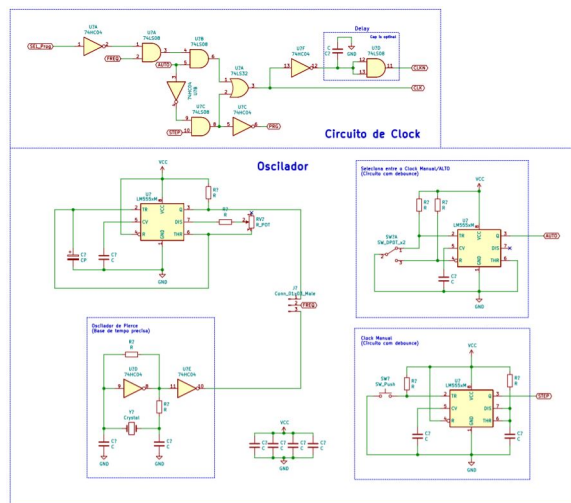


Fig. 5. Circuito de Clock e Oscilador

é utilizado para realizar as operações de soma e subtração e quatro portas NAND para a operação lógica. O resultado das operações é salvo no registrador para o ACC (74173) e um buffer (74244) permite a passagem dos valores do ACC para as portas de saída e para a realimentação da ULA. Um 74173 é utilizado para registrar os flags de carry e de zero. As portas de saída e os registradores são conectados a saída da ULA e são formados por circuitos 74173. O circuito do computador é composto por quatro portas de saída e por três registradores. Na Figura 7 pode se observar o circuito da Memória RAM e das portas de entradas. O circuito da Memória RAM foi implementada com o circuito integrado SRAM (6116) de 2k x 8. As instruções STW e LDW são responsáveis por gerarem os sinais para a manipulação dos dados de leitura/escrita na Memória RAM. As portas de entrada são dois buffers (74244), onde a saída está conectada com o barramento de dados.

III. RESULTADOS

Como resultado do presente trabalho, obteve-se uma melhoria de projeto de um computador de 4 bits implementado em um programa computacional gratuito e multiplataforma, com foco no ensino dos conteúdos sobre Eletrônica Digital, Microcontroladores e Microprocessadores. Com o conceito “Learn to Program and Program to use (LPPU)”, o computador não contém nenhum programa ou micro sistema gravado, com objetivo de estimular o estudante a entender seu funcionamento e o conjunto de instruções para que então possa desenvolver seus próprios programas. É oportuno mencionar que o projeto ainda está em desenvolvimento, e as próximas etapas, consistem no projeto de uma placa de circuito impresso e na redação de uma documentação detalhada que permita a qualquer estudante entender e programar o computador. O conjunto de instruções implementadas são:

- LDI: Carrega o imediato no ACC, RA, RB ou RD;
- NAND: Executa a operação NAND entre ACC e n, ACC e RA, ACC e RB ou ACC e RAM;

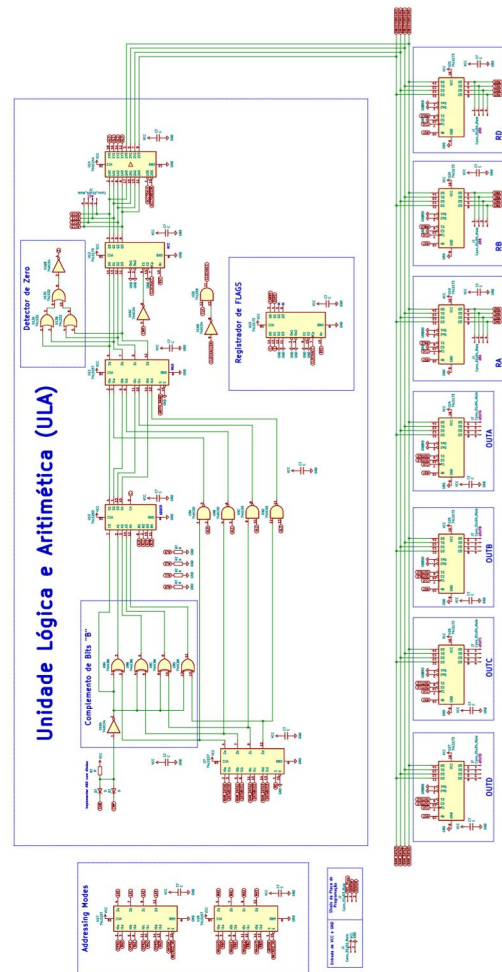


Fig. 6. Unidade Lógica e Aritmética (ULA), Registradores e Portas de Saída

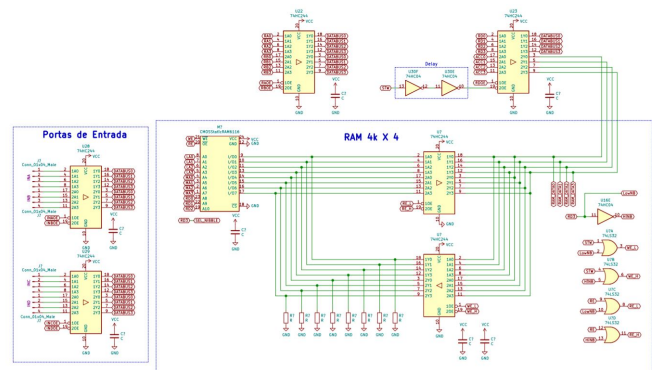


Fig. 7. Portas de Entrada e Memória RAM

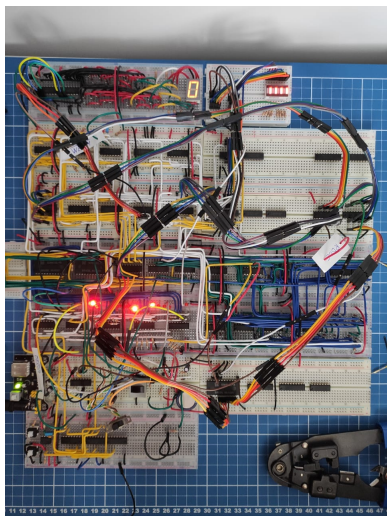


Fig. 8. Circuito Parcialmente Montado

- LDW: Carrega a RAM no ACC;
- LDA: Carrega RA em ACC, RB em ACC ou RD em ACC;
- OUTA: Coloca o imediato, o ACC, o RA ou a RAM em OUTA;
- OUTB: Coloca o imediato, o ACC, o RA ou a RAM em OUTB;
- OUTC: Coloca o imediato, o ACC, o RA ou a RAM em OUTC;
- LDR: Carrega RA com ACC, RB com ACC ou RD com ACC;
- CMP: Compara ACC com imediato, com RA, com RB ou com RAM;
- OUTD: Coloca o imediato, o ACC, o RA ou a RAM em OUTD;
- STW: Armazena ACC em RAM;
- SUB: Subtrai ACC com valor do imediato, com RA, com RB ou com RAM.
- JPI: Realiza o salto para o endereço RD:MA:LA
- JPC: Se CARRY = 1; Realiza o salto para o endereço MA:LA
- JPZ: Se ZERO = 1; Realiza o salto para o endereço MA:LA
- ADD: Soma ACC com valor do imediato, com RA, com RB ou com RAM.

Todas as instruções e suas variações com o nibble modificador foram verificadas por meio de pequenos programas, no qual obteve-se o resultado esperado. O circuito parcialmente montado é ilustrado na Figura 8.

IV. CONCLUSÃO

Com o presente projeto, os estudantes tiveram uma experiência única e a oportunidade de estudar todas as partes de um computador e consolidar conceitos fundamentais relacionados aos conteúdos que tratam da eletrônica digital, arquitetura de computadores, microcontroladores e microproces-

sadores. Com o computador, é possível realizar a inserção de programas diretamente em código de máquina, e permitindo o acompanhamento da execução passo a passo de cada instrução, aumentando o entendimento das etapas de funcionamento do computador e seus circuitos eletrônicos. Testes exaustivos foram realizados com todas as instruções a fim de verificar possíveis problemas no hardware do computador desenvolvido e verificou-se que em todos os casos, as instruções foram executadas corretamente.

AGRADECIMENTOS

Os autores agradecem ao Instituto Federal do Paraná (PIAP e PRADI), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (PBIC-Jr e PIBIC) e a Fundação Araucária (PIBITI) pelo apoio financeiro.

REFERENCES

- [1] TANENBAUM, Andrew S.; ZUCCHI, Wagner Luiz. Organização estruturada de computadores. Pearson Prentice Hall, 2013.
- [2] STALLINGS, William. Computer organization and architecture: designing for performance. Pearson, 2012.
- [3] FERLIN, Edson Pedro; PILLA, V. Driving the learning: microprocessors from theory to practice. In: Information Technology Based Proceedings of the Fifth International Conference on Higher Education and Training, 2004. ITHET 2004. IEEE, 2004. p. 459-462.
- [4] IBRAHIM, Dogan. A new approach for teaching microcontroller courses to undergraduate students. Procedia-Social and Behavioral Sciences, v. 131, p. 411-414, 2014.
- [5] PAPAZOGLU, Panayotis; MOSCHOS, Anastasios. OpenHardSim: An open source hardware-based simulator for learning microprocessors. In: 2017 IEEE Global Engineering Education Conference (EDUCON). IEEE, 2017. p. 157-160.
- [6] QARALLEH, Esam A.; DARABKH, Khalid A. A new method for teaching microprocessors course using emulation. Computer Applications in Engineering Education, v. 23, n. 3, p. 455-463, 2015.
- [7] MANSEUR, Rachid. Studio teaching: An embedded systems course. In: 2018 IEEE Global Engineering Education Conference (EDUCON). IEEE, 2018. p. 590-598.
- [8] BLACK, Michael. Export to arduino: a tool to teach processor design on real hardware. Journal of Computing Sciences in Colleges, 2016.
- [9] ZHANG, Wei. Teaching the Introductory Computer Architecture Course with a Systematic View. Proceedings of the 2007 Midwest Section Conference of the American Society for Engineering Education. 2007.
- [10] GHANI, M. A. H. A.; ENZAI, N. I. M.; AHMAD, N. Effect of using simulator as teaching aid on microprocessor course performance. In: 2017 IEEE 15th Student Conference on Research and Development (SCORED). IEEE, 2017. p. 215-219.
- [11] FEINBERG, Dave. A simple and affordable TTL processor for the classroom. Computer Science Education, v. 17, n. 2, p. 107-116, 2007.
- [12] SANTA, Fernando Martinez; SARMIENTO, Fredy H. Martinez; ARIZA, Holman Montiel. Minimalist 4-bit Processor Focused on Processors Theory Teaching. Indian Journal of Science and Technology, v. 10, n. 14, p. 1-6, 2017.
- [13] KOSTADINOV, Nikolay; BENCHEVA, Nina. An Approach for Teaching Processor Design. In: 2018 28th EAEEIE Annual Conference (EAEEIE). IEEE, 2018. p. 1-9.
- [14] KOSTADINOV, Nikolay; BENCHEVA, Nina. An Approach for Teaching Processor Design and How to Extend its Features. In: 2019 29th Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE). IEEE, 2019. p. 1-4.
- [15] ALOMAR, Alejandro. Computadora de 4 BITS. 2018. Disponível em: <http://www.sinaptec.alomar.com.ar/2018/03/computadora-de-4-bits-capitulo-1.html>. Acesso em: 07 mar. 2020.
- [16] Tasso, Matheus Fernando; Brito, Lucas Fregonele; Paula, João Lucas Pereira dos Santos; Nunes, Carlos Daniel de Souza; Acordi, Edson Junior. Proposal of a didactic computer for teaching digital electronics and microprocessor concepts. In: 2020 V Simpósio de Controle e Processos Industriais (SIMPROIM).