

APS-4: Nível do Sistema de Computador

Jogar

Jogue o jogo NAND: <https://nandgame.com/>

O portão NAND serve como uma pedra angular na lógica digital, sendo universalmente capaz de construir todas as funções lógicas por meio de combinações inteligentes. Neste jogo cativante, você embarcará em uma jornada de descoberta, construindo progressivamente funções lógicas cada vez mais intrincadas, desde o rudimentar até o sistêmico. Cada nível se baseia no fundamento do anterior, reforçando sua compreensão enquanto você navega pelos desafios. Esta empreitada lúdica, porém educativa, oferece um vislumbre prático da aplicação de conceitos teóricos discutidos em aulas, aproximando a teoria da prática na construção de sistemas de computadores.

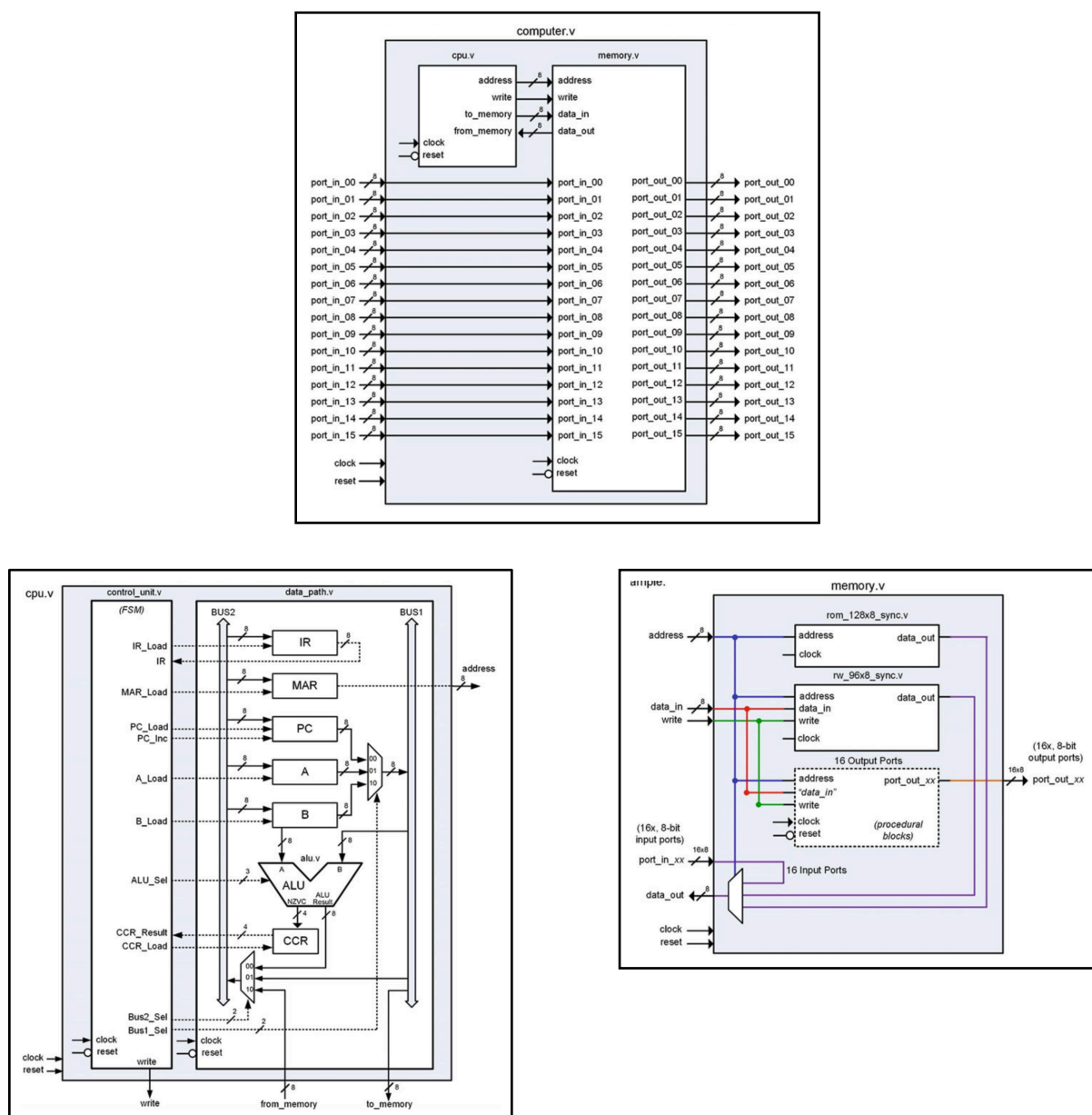
1. **(1.6 pontos)** Complete o nível cinco: "Processor". Documente seu progresso por meio de capturas de tela do seu design e uma explicação por escrito.

Desenvolver

Os processos de design contemporâneos dependem fortemente de ferramentas de design assistido por computador (CAD) para gerenciar a escala e a complexidade das arquiteturas digitais modernas. No cerne dessa metodologia estão as linguagens de descrição de hardware (HDLs) como Verilog. Por meio das HDLs, modelos descritivos de componentes de hardware são elaborados, possibilitando simulações rigorosas para validação de modelos e facilitando a síntese e minimização assistidas por computador para a realização da circuitaria final. Dominar tais ferramentas requer uma jornada semelhante à empreendida no jogo do NAND - partindo do básico e progredindo gradualmente até o nível do sistema, enquanto aprimora suas habilidades e entendimento.

Objetivo: Implemente um sistema de computador de 8 bits `computer.v`. O sistema está estruturado em dois níveis hierárquicos mostrados na Fig. 1. O arquiteto especificou templates para os blocos principais: `computer.v`, `memory.v`, `data_path.v` e `control_unit.v`, que precisam ser complementados pelas equipes de desenvolvedores. Você pode utilizar todos os módulos desenvolvidos nas atividades anteriores (APS1.2, APS2.2 e APS3.2), especialmente, `ALUb.v`, `reg8b.v` e `count8b.v`. Para documentar seu projeto você precisa escrever um relatório do projeto (arquivo .pdf) e fornecer todos os módulos Verilog desenvolvidos e seus testbenches.

Fig. 1: Diagrama de blocos do sistema de computador de 8 bits



2. **(2 pontos)** *Memory:* Complete o módulo de **memory.v**. O arquiteto forneceu módulos adicionais (**rom_128x8_sync.v**, **rw_96x8_sync.v**, **port_out8_sync.v**) que podem ser usados como estão. Valide seu projeto com uma bancada de testes adequada cobrindo casos de uso essenciais, por exemplo, (i) Os dados podem ser armazenados de **data_in** na memória (ROM, RAM, Saída) dependendo do endereço; (ii) Os dados podem ser enviados para **data_out** da memória (ROM, RAM) ou portas de entrada; (iii) As portas de saída são mapeadas diretamente para a memória IO, etc. Você precisa incluir capturas de tela das formas de onda simuladas em seu relatório.

3. **(2 pontos)** *Data Path*: Conclua o módulo **data_path.v**. Valide seu projeto com uma bancada de testes adequada cobrindo casos de uso essenciais, por exemplo, (i) Registros (IR, PC, A, B) podem ser carregados from_memory com um novo valor; (ii) PC, A ou B podem ser enviados para_memory dependendo de Bus1_Sel; (iii) A ALU pode realizar diferentes operações e os resultados monitorados em CCR_Result e to_memory; (iv) MAR pode ser carregado from_memory com um novo valor visível no endereço, etc. Você precisa incluir capturas de tela das formas de onda simuladas em seu relatório.
4. **(2 pontos)** *Controller*: Conclua o módulo **control_unit.v**. O modelo do arquiteto segue a abordagem de três blocos (STATE_MEMORY, NEXT_STATE_LOGIC, OUTPUT_LOGIC). Sintetize seu código pelo menos para um número selecionado de instruções. Leia os códigos hexadecimais (IR) das instruções correspondentes e verifique as saídas de controle. Você precisa incluir capturas de tela do esquema e dos casos de uso em seu relatório.
5. **(2 pontos)** *Run a Program*: Conclua o módulo **computador.v**. Escrever um programa simples em alto nível e traduzir para uma sequência de instruções de máquina. Carregue o conjunto de instruções na ROM (veja rom_128x8_sync.v para obter um exemplo). Verifique a execução correta com uma bancada de testes adequada. Você precisa documentar a execução no relatório .pdf (i) Projetando diagramas de estado de instruções selecionadas do seu programa e (ii) Incluindo e explicando as formas de onda simuladas para as mesmas instruções (veja também Aula 6-4).
6. **(0.4 pontos)** Escreva uma autoavaliação muito curta sobre o sucesso do projeto. O controlador funciona conforme o esperado ou ainda apresenta alguns erros a serem corrigidos? O que sua equipe poderia ter feito melhor em seu próximo projeto digital? Dê uma pontuação ao seu projeto (n de 10).