

Por diferentes razões tais como simplicidade de programação, portabilidade e tamanho de código reduzido, Python se tornou uma das linguagens de programação mais expressivas atualmente, até mesmo em comunidades como a de sistemas embarcados. Todavia, não há indícios de existência de uma máquina em hardware que execute os bytecodes Python nativamente possibilitando o aumento expressivo da eficiência de execução. Sendo assim, nosso intuito é descrever em VHDL um processador Python que suporte um subconjunto de instruções da PVM (Python Virtual Machine) distribuídas em quatro classes principais de operação: instruções aritméticas, controle de fluxo, acesso à memória e suporte a subrotinas. Uma vez que não existe até o momento qualquer especificação referente a arquitetura da PVM que possa nortear projetistas no desenvolvimento de máquinas Python, nossa compreensão sobre a estrutura e comportamento da PVM foi produzida através da leitura de descrições incompletas e superficiais existentes sobre o conjunto de instruções da PVM, assim como através do estudo do compilador Python e do código fonte da PVM. Pensando em construir uma arquitetura com complexidade e eficiência intermediárias, optamos por desenvolver o projeto de um multiciclo. Além disso, uma vez que a PVM realiza dinamicamente a resolução de tipos dos dados, optamos por construir inicialmente uma arquitetura com suporte estritamente a dados do tipo inteiro, enquanto estudamos alternativas para a implementação do suporte para resolução dinâmica que tenham baixo impacto, em termos de recursos (tais como área do hardware, memória e energia) e velocidade, sobre a arquitetura já planejada. Espera-se que, ao final deste trabalho, características intrínsecas da PVM, aliadas às nossas decisões de projeto possibilitem a obtenção de resultados que relevem potenciais vantagens do uso deste processador em sistemas embarcados.