PCS3335 - Laboratório Digital A - Experiência 7

por Bruno de Carvalho Albertini

15/04/2024

A experiência 7 junta as experiências 4, 5 e 6, adicionando comunicação ao `multisteps`.

Introdução

O módulo da experiência 4, o `multisteps` é a base do algoritmo SHA256 que estamos desenvolvendo. No entanto, sem uma comunicação externa o módulo não pode ser usado na prática. As experiências 5 e 6 são a transmissão e a recepção serial, respectivamente.

Na experiência 7, você deve montar um módulo chamado `sha256_1b` que segue a assinatura da Figura 1. O seu módulo recebe uma palavra de 1B pela serial e calcula o *hash* usando o módulo `multisteps`. A cada palavra recebida, você deve reiniciar o cálculo com o *byte* recebido, mesmo que o cálculo anterior não tenha acabado. Caso você não receba nenhum valor por um número de ciclos de *clock* serial suficientes, você terminará o cálculo (e.g. 64 ciclos de serial com uma tolerância de alguns ciclos). No caso de o `multisteps` terminar o cálculo, você deve transmitir o *byte* menos significativo do *hash* calculado pela serial.

Experiência 5

Seu objetivo nesta experiência montar o módulo que junta as três experiências com o comportamento descrito acima.

Planejamento

Para o juiz, você deve seguir a assinatura exata do módulo, conforme Figura 1.

```
entity sha256_1b is
  port (
    clock, reset : in bit;
    serial_in: in bit;
    serial_out: out bit
  );
end sha256_1b;
```

Figura 1: Entidade para a experiência 7

O *clock* é o da placa (50MHz) e qualquer divisão deve ser feita por você. O *reset* é assíncrono ativo alto e coloca o sistema em estado de espera pelo primeiro byte serial. A entrada e saída serial tem comportamento similar às experiências 5 e 6.

Note que é possível que o usuário envie um *byte* a qualquer momento, o que significa que você deve cancelar qualquer operação de *hash* em andamento e reiniciar o cálculo com o último *byte* recebido, mesmo sem um *reset*. O valor de retorno só acontece se a linha serial permanecer em repouso ao menos por 64 ciclos de *clock*, e é enviado apenas uma vez.

Preparação para montagem e Execução

Para a montagem e execução, sugerimos uma adição no seu módulo, mostrando de alguma maneira o que está sendo recebido, o estado do `multisteps`, e o que está sendo enviado. Ainda será necessário o *reset*, que deve obrigatoriamente vir da Analog Discovery, assim como a transmissão e recepção serial (use o módulo Protocol).

As adições não são obrigatórias, porém facilitam a avaliação.

Desafio

Para esta experiência, o desafio vale 20% da nota, será divulgado na hora pelo seu professor e deverá ser implementado em sala. Caso pretenda fazer o desafio, sugerimos que use os horários de *open lab* para testar sue projeto antes da aula, permitindo assim que o tempo em sala seja dispendido com o desafio.

Dicas

- Planeje sua FSM no papel antes de implementar e inclua uma foto/-desenho do seu diagrama no planejamento para facilitar os testes.
- O *reset* é assíncrono, ou seja, seu circuito não retornará nada se um *reset* for feito a qualquer momento antes do final de um cálculo.
- Use a visualização em hexadecimal no terminal de recepção serial do Analog Discovery pois nem todos os códigos de 8b representam um caractere ASCII válido.
- Lembre-se que o `multisteps` recebe na verdade 512b, que são a replicação nos 8b recebidos pela serial 64 vezes. Não replique dentro do `multisteps`.
- As mesmas tabelas da experiência 4 (no fórum) podem ser usadas para validar o módulo.