

Trabalho 2

FIFO Assíncrona

Entrega:

Entrega
17/09

Objetivo Geral

Implementação e análise dos impactos no processo de síntese lógica e timing no emprego de circuitos digitais com características sequenciais síncronas e/ou assíncronas.

Especificação:

Desenvolva um circuito digital, em VHDL, que possua o comportamento funcional de uma FIFO (First In First Out) assíncrona cujas características funcionais e estruturais são apresentadas a seguir. A Figura 1 e a Tabela 1 apresentam as interfaces e suas respectivas funções no bloco lógico.

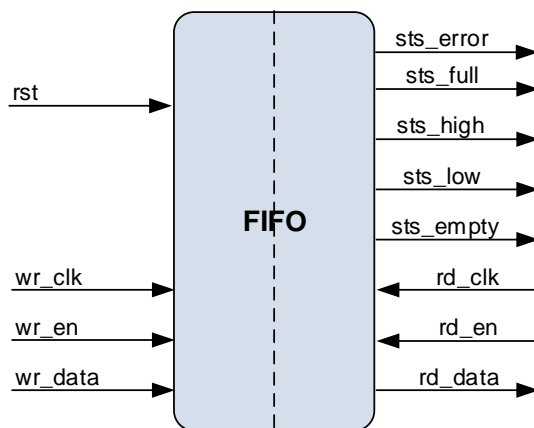


Figura 1 - Interfaces do Bloco.

Nome	Tipo	Largura	Função
rst	Input	1 bit	Reset assíncrono.
wr_clk	Input	1 bit	Referência de sincronismo de escrita.
wr_en	Input	1 bit	Habilitação de escrita.
wr_data	Input	8 bits	Barramento de dado (escrita).
rd_clk	Input	1 bit	Referência de sincronismo de leitura.
rd_en	Input	1 bit	Habilitação de leitura.
rd_data	Output	8 bits	Barramento de dado (leitura).
sts_full	Output	1 bit	Status de FIFO cheia.
sts_empty	Output	1 bit	Status de FIFO vazia.
sts_high	Output	1 bit	Status de FIFO quase cheia.
sts_low	Output	1 bit	Status de FIFO quase vazia.
sts_error	Output	1 bit	Status de erro.

Tabela 1 - Interfaces da Entidade.

- A entidade deverá ser nomeada como FIFO_ASYNC e deverá ser totalmente sintetizável.
- A FIFO deverá possuir a capacidade de armazenar 64 posições de dados, cada uma palavra de 8 bits;
- O circuito deverá possuir um reset (**rst**) ativo em nível lógico alto, e característica assíncrona;
- Quando o sinal de reset (**rst**) for aplicado, todas as palavras de dado da FIFO deverão ser sobrescritas com o valor zero (x"00");

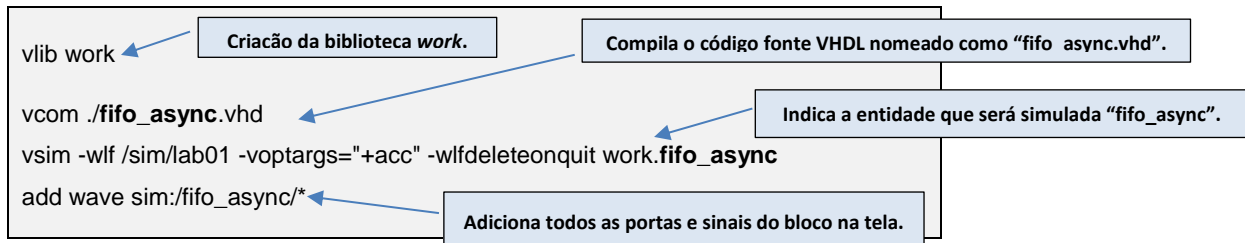
- O processo de escrita na FIFO deverá ser sensível a borda de subida do sinal de clock relativo a escrita (**wr_clk**);
- O processo de leitura na FIFO deverá ser sensível a borda de subida do sinal de clock relativo a leitura (**rd_clk**);
- O processo de escrita na FIFO deverá ocorrer somente quando a entrada de enable (**wr_en**) estiver em nível lógico alto;
- O processo de leitura na FIFO deverá ocorrer somente quando a entrada de enable (**rd_en**) estiver em nível lógico alto;
- O sinal de **sts_full** deverá ser colocado em nível lógico alto (e assim permanecer) quando a FIFO estiver completamente cheia;
- O sinal de **sts_empty** deverá ser colocado em nível lógico alto (e assim permanecer) quando a FIFO estiver completamente vazia;
- O sinal de **sts_high** deverá ser colocado em nível lógico alto (e assim permanecer) quando a FIFO estiver prestes a ficar completamente cheia, isto é, quando faltarem 4 (ou menos) posições de memória para serem preenchidas por escritas;
- O sinal de **sts_low** deverá ser colocado em nível lógico alto (e assim permanecer) quando a FIFO estiver prestes a ficar completamente vazia, isto é, quando faltarem 4 (ou menos) posições de memória para serem lidas;
- O sinal de **sts_error** deverá ser colocado em nível lógico alto (e assim permanecer) quando a FIFO tiver um overflow dos ponteiros, isto é, quando ela sobrescrever uma posição que ainda não foi lida, ou ler uma posição já lida;
- Após o sinal de **sts_error** for colocado em nível lógico alto por indicação de algum erro, este só deve ser colocado em nível lógico baixo após um reset global da FIFO;

Simulação – Modelsim:

- Acessando via SSH a paxos, você deverá carregar o módulo do Modelsim no Terminal aberto.

```
source /soft64/source_gaph
module load modelsim
vsim &
```

- No Transcript do Modelsim



Síntese – Cadence Genus:

- Acessando via SSH a paxos, você deverá carregar o módulo do Genus no Terminal aberto.

```
source /soft64/source_gaph
module load genus
genus
```

- Definição da biblioteca que será utilizada neste projeto.

```
set_db library /soft64/design-kits/stm/65nm-cmos065_536/CORE65GPSVT_5.1/libs/CORE65GPSVT_nom_1.00V_25C.lib
```

- Leitura do(s) arquivo(s) VHDL que compõe o projeto. Neste exemplo o código fonte foi nomeado como *"fifo_sync.vhd"*

```
read_hdl -vhdl fifo_async.vhd
```

- Elaboração do projeto. Neste exemplo a entidade do projeto foi nomeada como *"fifo_sync"*.

```
elaborate fifo_async
```

- Síntese Lógica para Células Genéricas:

```
syn_generic
```

- Síntese Lógica para células da biblioteca alvo do projeto:

```
syn_map
```