

Aplicação da ULA

Autores

Gabriel A. F. Souza, Gustavo D. Colletta, Leonardo B. Zoccal, Odilon O. Dutra

Unifei

Histórico de Revisões

27 de janeiro de 2025	1.0	Primeira versão do documento.
-----------------------	-----	-------------------------------

Tópicos

- Datapath de microcontrolador
- Exercícios

Datapath de microcontrolador

Circuito completo

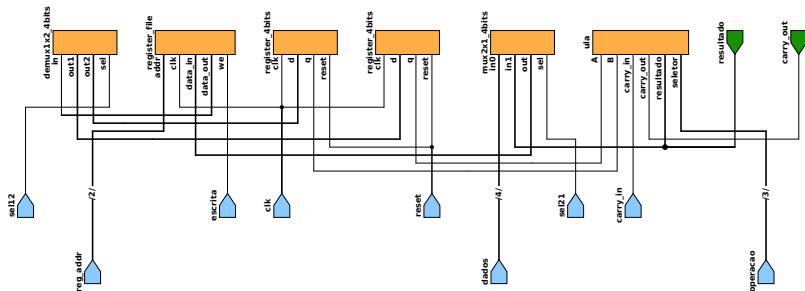


Figura 1: Diagrama RTL do *datapath* de 4-bits.

Circuito completo

① Entradas:

- dados[3:0]: Entrada de 4 bits que pode ser usada para alimentar os registros ou ser passada pela ULA.
- sel21: Sinal de controle para o **mux 2x1**. Ele seleciona entre os dados de entrada e o resultado da ULA.
- sel12: Sinal de controle para o **demux 1x2**. Ele decide qual saída do **register_file** vai ser alimentada na ULA.
- clk: Sinal de clock para o **register_file** e sincronização.
- operacao[2:0]: Sinal de 3 bits para selecionar a operação da ULA (AND, OR, soma, etc.).

Circuito completo

② Saídas:

- `resultado[3:0]`: Resultado de 4 bits da operação realizada pela ULA.
- `carry_out`: Carry-out da ULA, indicando se houve um carry gerado ou um borrow na operação aritmética.

Circuito completo

③ Componentes:

- **mux2x1_4bits**: O multiplexador seleciona entre os dados de entrada (dados) ou o resultado da ULA para alimentar o **register_file**.
- **register_file**: Armazena o valor da entrada e o disponibiliza para a ULA, quando necessário.
- **demux1x2_4bits**: Direciona a saída do **register_file** para as entradas da ULA.
- **ula**: Realiza operações aritméticas e lógicas (como soma, subtração, AND, OR, etc.) usando as entradas A e B e gerando o resultado e o **carry_out**.

Circuito completo

④ Funcionamento:

- O **mux2x1** escolhe entre o valor de entrada dados e o resultado da ULA, com base no sinal **sel21**, e envia o valor para o **register_file**.
- O **register_file** armazena o valor recebido e, com base no sinal **sel12**, o **demux1x2** direciona as saídas dos registros para as entradas A e B da ULA.
- A **ULA** realiza a operação especificada por operacao nos dados fornecidos pelas saídas do **demux**, e o resultado é enviado de volta como resultado, com o **carry_out** também gerado.

Register file

- **clk**: Sinal de clock utilizado para sincronizar as operações de escrita.
- **we**: Sinal de habilitação para escrita (**Write Enable**). Quando ativo, permite que o valor em `data_in` seja gravado no registro selecionado pelo endereço.
- **addr**: Endereço de 2 bits que seleciona um dos 4 registros (endereço único para cada registro).
- **data_in**: Dados de entrada (4 bits) que serão gravados no registro selecionado durante a escrita.
- **data_out**: Dados de saída (4 bits) do registro selecionado. Esse valor reflete o conteúdo do registro correspondente ao endereço fornecido.

Register file

❶ Seleção do Registro:

O endereço de 2 bits (`addr`) determina qual dos 4 registros está ativo. Apenas o registro correspondente ao endereço recebe os dados ou fornece os dados para `data_out`.

❷ Escrita:

Quando o sinal `we` está ativo (1) e ocorre uma borda de subida no clock (`clk`), os dados presentes em `data_in` são gravados no registro selecionado pelo endereço `addr`.

❸ Leitura:

O conteúdo do registro selecionado pelo endereço `addr` é direcionado para a saída `data_out`.

Multiplexador 2x1

O multiplexador 2x1 de 4 bits seleciona um entre dois vetores de entrada de 4 bits (`in0` e `in1`) com base no valor de um sinal de seleção (`sel`). A saída é um vetor de 4 bits correspondente à entrada selecionada.

① Entradas

- `in0`: Entrada 0 (4 bits).
- `in1`: Entrada 1 (4 bits).
- `sel`: Sinal de seleção (1 bit):
 - `sel` = 0: A saída será igual a `in0`.
 - `sel` = 1: A saída será igual a `in1`.

Multiplexador 2x1

② Saída

- out: Vetor de 4 bits contendo o valor da entrada selecionada.

③ Lógica

A lógica é implementada usando o operador ternário (`? :`):

- `out = (sel) ? in1 : in0.`
- Se `sel` for igual a 1, a saída será igual a `in1`. Caso contrário, a saída será igual a `in0`.

Demux 1x2

- **Entradas:**

- `in[3:0]`: Entrada de 4 bits.
- `sel`: Sinal de seleção de 1 bit que decide para qual saída a entrada será direcionada.

- **Saídas:**

- `out0[3:0]` e `out1[3:0]`: As saídas de 4 bits. Somente uma delas será ativada com o valor da entrada, dependendo do valor de `sel`.
- Quando `sel` for 0, a entrada `in` é passada para a saída `out0`.
- Quando `sel` for 1, a entrada `in` é passada para a saída `out1`.

registro de 4 bits

- **Entradas:**

- **clk**: Sinal de **clock**. O valor do registrador é atualizado na borda de subida do clock.
- **reset**: Sinal de **reset**. Quando está em nível alto (1), o registrador é resetado para 4'b0000.
- **d[3:0]**: A entrada de dados de 4 bits que será armazenada no registrador.

- **Saída:**

- **q[3:0]**: A saída de 4 bits, que armazena o valor atual do registrador.

registro de 4 bits

- **Comportamento:**

- O **registrador** armazena o valor de entrada `d` na borda de subida do sinal `clk`.
- Se o sinal `reset` for ativado (em nível alto), o registrador é resetado para `4'b0000` independentemente do valor de `clk`.
- Se `reset` não estiver ativo, o valor de `d` é transferido para `q` na borda de subida do `clk`.

Exercícios

Exercício Único

- 1 Utilizando os módulos fornecidos, implemente o *datapath* de 4-bits estudado.
- 2 Escreva um arquivo de *testbench* capaz de testar o módulo projetado.