

Aluno: \_\_\_\_\_

Matrícula: \_\_\_\_\_ Data: \_\_\_\_\_

### Sprint 8 – Entrada e saída paralela – CPU MIPS

**Descrição geral do problema:** Incluir uma entrada e uma saída paralela mapeada em memória. Isso finalizará o conjunto mínimo de funcionalidades da CPU.

#### Requisitos mínimos:

Abra o projeto da Sprint7 e edite-o para incluir as funcionalidades dessa sprint. **Obs: “File > Open Project” e NÃO “File > Open”.**

- Até esse momento, a CPU v0.3 não tinha nenhuma forma de trocar dados com o mundo externo. Além das interfaces de debug, a única entrada externa da montagem era o clock. A fim de completar a versão v1.0 da CPU, inclua uma porta de entrada e uma de saída paralela, mapeada em memória.
  - O endereço 8'hFF da memória de dados será inutilizado e ressignificado para as portas de entrada e saída paralelas, mapeadas em memória.
  - Ao armazenar o conteúdo de algum registrador **\$X** no endereço 8'hFF da memória de dados, **SW \$X, FF(\$0)**, o bloco **ParallelOUT** redirecionará o conteúdo de **\$X** para a saída paralela **w\_DataOut**. A especificação lógica do circuito de saída está ilustrada na Figura 2
  - Ao carregar o conteúdo do endereço 8'hFF da memória de dados, para um registrador **\$X**, **LW \$X, FF(\$0)**, o bloco **ParallelIN** redirecionará o conteúdo da entrada paralela **w\_DataIn** para o registrador **\$X**. A especificação lógica do circuito de saída está ilustrada na Figura 3
  - A sugestão de montagem final da CPU v1.0 está representada na Figura 1.

Perceba que não foi necessário criar mais nenhuma instrução para manipular as portas. Somente **SW** e **LW**

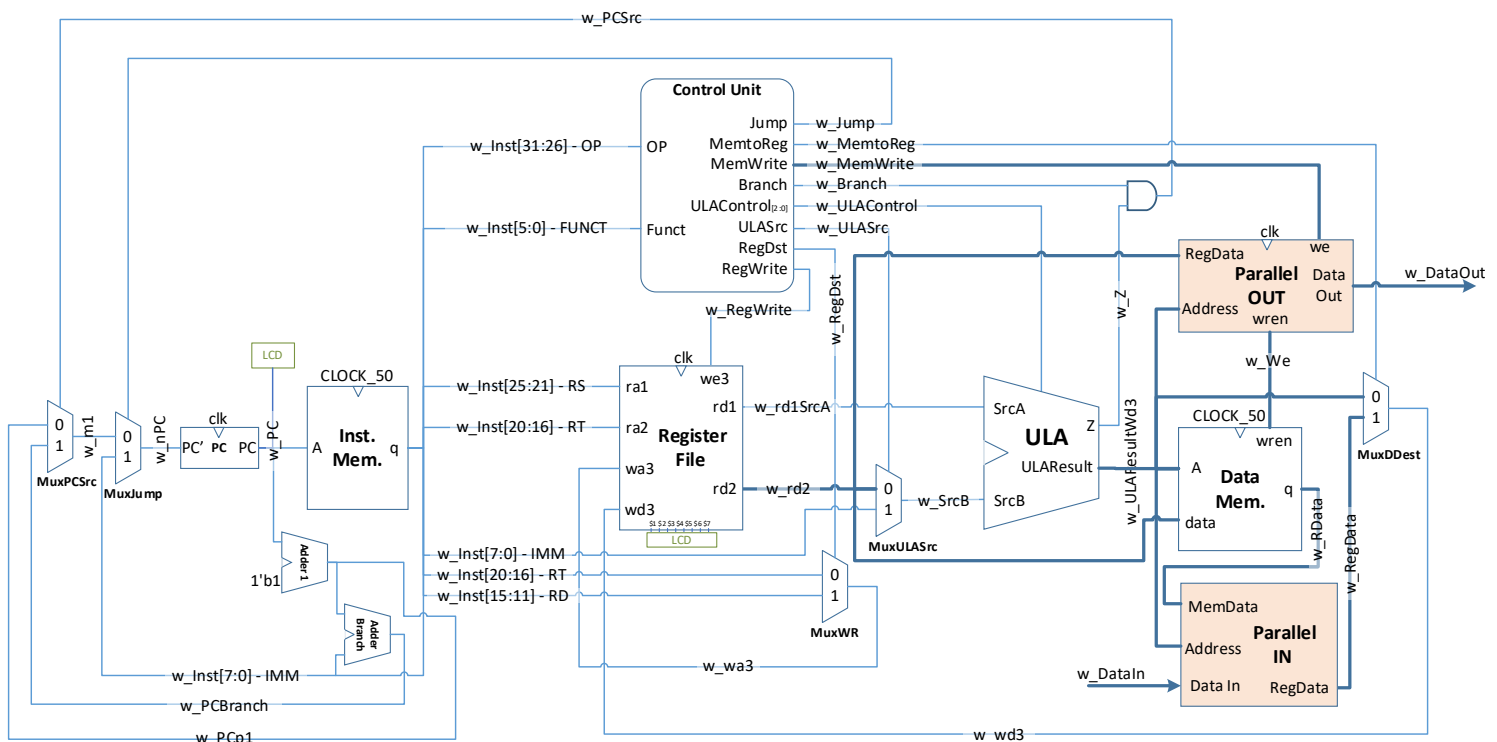


Figura 1 – CPU V1.0 e memórias

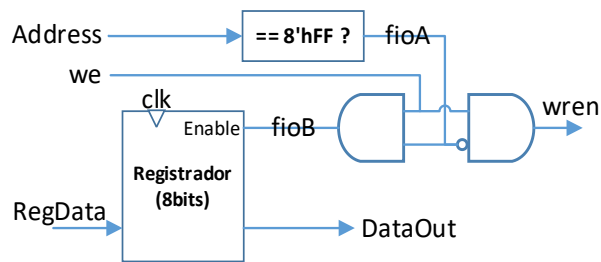


Figura 2 – Saída paralela mapeada no endereço 8'hFF da memória

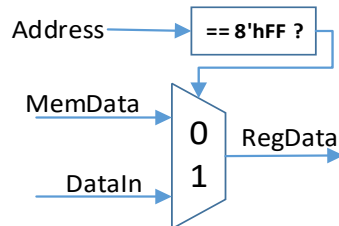


Figura 3 – Entrada paralela mapeada no endereço 8'hFF da memória

2. Ligações auxiliares para Debug:

- Conecte a saída paralela (DataOut) no display **w\_d1x4**,
- Conecte a entrada paralela (DataIn) nas chaves **SW[7:0]**

3. Para testar sua montagem, escreva e rode na sua CPU, um programa em assembly que receba um número de 8bits na entrada paralela. Retorne, na saída paralela, ZERO se o número for par e UM se o número for ímpar. Aumente o clock principal da CPU para 10Hz.

Relembrando o conjunto de instruções suportadas pela CPU

Instrução	Descrição	Algoritmo
ADD \$X, \$Y, \$Z	Adicionar	$\$X = \$Y + \$Z$
SUB \$X, \$Y, \$Z	Subtrair	$\$X = \$Y - \$Z$
AND \$X, \$Y, \$Z	AND Bit a bit	$\$X = \$Y \& \$Z$
OR \$X, \$Y, \$Z	OR Bit a bit	$\$X = \$Y   \$Z$
SLT \$X, \$Y, \$Z	Menor que	$\$X = 1$ se $\$Y < \$Z$ e 0 c.c.
LW \$X, i(\$Y)	Carregar da memória	$\$X \leftarrow \text{Cont. do end. } (\$Y + i)$
SW \$X, i(\$Y)	Armazenar na memória	$\text{End. } (\$Y + i) \leftarrow \$X$
BEQ \$X, \$Y, i	Desviar se igual	Se $\$X == \$Y$ , $PC = PC + 1 + i$
ADDi \$X, \$Y, i	Adicionar Imediato	$\$X = \$Y + i$
J i	Desvio incondicional	$PC = i$

Tabela 1 –Conjunto de instruções MIPS suportadas pela CPU do LASD

**Desafio (Valendo +0,1 na média geral)**

- Conecte o display de 7 segmentos HEX0 na saída paralela da sua CPU
- Escreva um programa, em assembly, para acender e apagar os segmentos da periferia do display, em sequência, um por vez, dando a ilusão de movimento circular.
- Quando for recebido o valor ZERO na entrada paralela, os segmentos devem girar no sentido horário, caso seja recebido UM, os segmentos devem girar no sentido anti-horário