



Universidade Federal
de Campina Grande

Centro de Engenharia Elétrica e Informática
Disciplina: Lab. de Arquitetura de Sistemas Digitais
Aluna: Nicole Duarte De Melo
Matrícula: 120110393

Sprint 10

Projeto de um Jogo: Pedra Papel Tesoura

O jogo possui dois jogadores A e B e cada combinação da entrada paralela diz se um jogador ganhou ou não ou se foi um empate.

Na parte superior do display de LCD mostra a posição dos dois jogadores; Na parte debaixo é mostrado o placar de cada um, seguido pelo placar de empate, denotado por "E: ".

Considerando a entrada paralela FF(\$0) possuindo a sequência de 8 bits:

$$Input = B_7 B_6 B_5 B_4 B_3 B_2 B_1 B_0$$

- Possibilidades de Jogada para o Player B:

- Player B jogou Pedra: $B_0 = 1$;
- Player B jogou Papel: $B_1 = 1$;
- Player B jogou Tesoura: $B_2 = 1$;

- Possibilidades de Jogada para o Player A:

- Player A jogou Pedra: $B_3 = 1$;
- Player A jogou Papel: $B_4 = 1$;
- Player A jogou Tesoura: $B_5 = 1$;

O bit B_7 funciona como uma chave de habilitação; Quando $B_7 = 1$ a combinação de entradas dos jogadores A e B é lida e vai para o placar no display de LCD.

- Possibilidades e entradas para cada jogada:

- **Empate:** A = Pedra e B = Pedra: $Input = 10001001 = 89'h$
- **Empate:** A = Papel e B = Papel: $Input = 10010010 = 92'h$
- **Empate:** A = Tesoura e B = Tesoura: $Input = 10100100 = A4'h$
- **"B" ganha:** A = Tesoura e B = Pedra: $Input = 10100001 = A1'h$
- **"B" ganha:** A = Pedra e B = Papel: $Input = 10001010 = 8A'h$
- **"B" ganha:** A = Papel e B = Tesoura: $Input = 10010100 = 94'h$
- **"A" ganha:** A = Papel e B = Pedra: $Input = 10010001 = 91'h$
- **"A" ganha:** A = Tesoura e B = Papel: $Input = 10100010 = A2'h$
- **"A" ganha:** A = Pedra e B = Tesoura: $Input = 10001100 = 8C'h$

As chaves SW[0] recebem os bits da entrada input, seguindo a sequência de significância dos bits.

Novas Funcionalidades implementadas na CPU

1. O número de registradores aumentou de 8 para 16 registradores de 8 bits;
2. Novas instruções foram implementadas na CPU, tais como: ANDi, ORi, SLTi e BNE;
3. Foram utilizadas as novas instruções no código em Assembly para implementação do projeto e todos os registradores criados;
4. Foi alterado o código para mapeamento do display de LCD. Nessa nova forma cada partição recebe um valor que é convertido em código ASCII para melhor visualização do placar do jogo;
5. O código permite a habilitação ou não das jogadas, com uma chave de habilitação que só adiciona pontos a uma jogada caso seja colocada uma das entradas especificadas. Caso não seja colocada uma entrada válida, o placar não altera e o jogo prossegue até que seja posta uma das possíveis entradas e a chave de habilitação for acionada.

- Código Assembly comentado:

LW \$1, FF(\$0)	//Lê a entrada paralela e joga para o registrador \$1;
SLTi \$2, \$1, 80	//Se \$1 < 80'h \$2 = 1, \$2 = 0 c.c.
BNE \$0, \$2, FD	//Se \$2 != 0, sobe duas linhas, caso o contrário ele continua o código;
ADDi \$3, \$0, 89	//\$3 = 89'h - A = Pedra e B = Pedra;
ADDi \$4, \$0, 92	//\$4 = 92'h - A = Papel e B = Papel;
ADDi \$5, \$0, A4	//\$5 = A4'h - A = Tesoura e B = Tesoura;
ADDi \$6, \$0, A1	//\$6 = A1'h - A = Tesoura e B = Pedra;
ADDi \$7, \$0, 8A	//\$7 = 8A'h - A = Pedra e B = Papel;
ADDi \$8, \$0, 94	//\$8 = 94'h - A = Papel e B = Tesoura;
ADDi \$9, \$0, 91	//\$9 = 91'h - A = Papel e B = Pedra
ADDi \$A, \$0, A2	//\$A = A2'h - A = Tesoura e B = Papel;
ADDi \$B, \$0, 8C	//\$B = 8C'h - A = Pedra e B = Tesoura;
BNE \$1, \$3, 2	/*Compara a entrada "\$1" com uma possibilidade de empate no "\$3"
	Se \$3 = \$1, segue o código, caso o contrário pula 2 linhas*/
ADDi \$F, \$F, 1	//\$F = \$F + 1 - Soma no placar de empate que está no registrador \$F;
J 26	//Pula para a linha 26;
BNE \$1, \$4, 2	/*Compara a entrada "\$1" com uma possibilidade de empate no "\$4"
	Se \$4 = \$1, segue o código, caso o contrário pula 2 linhas*/
ADDi \$F, \$F, 1	//\$F = \$F + 1 - Soma no placar de empate que está no registrador \$F;
J 26	//Pula para a linha 26;
BNE \$1, \$5, 2	/*Compara a entrada "\$1" com uma possibilidade de empate no "\$5"
	Se \$5 = \$1, segue o código, caso o contrário pula 2 linhas*/
ADDi \$F, \$F, 1	//\$F = \$F + 1 - Soma no placar de empate que está no registrador \$F;
J 26	//Pula para a linha 26;
BNE \$1, \$6, 2	/*Compara a entrada "\$1" com uma possibilidade de "B ganhar no "\$6"
	Se \$6 = \$1, segue o código, caso o contrário pula 2 linhas*/
ADDi \$D, \$D, 1	//\$D = \$D + 1 - Soma no placar de "B" que está no registrador \$D;
J 26	//Pula para a linha 26;
BNE \$1, \$7, 2	/*Compara a entrada "\$1" com uma possibilidade de "B ganhar no "\$7"
	Se \$7 = \$1, segue o código, caso o contrário pula 2 linhas*/
ADDi \$D, \$D, 1	//\$D = \$D + 1 - Soma no placar de "B" que está no registrador \$D;
J 26	//Pula para a linha 26;
BNE \$1, \$8, 2	/*Compara a entrada "\$1" com uma possibilidade de "B ganhar no "\$8"
	Se \$8 = \$1, segue o código, caso o contrário pula 2 linhas*/
ADDi \$D, \$D, 1	//\$D = \$D + 1 - Soma no placar de "B" que está no registrador \$D;
J 26	//Pula para a linha 26;
BNE \$1, \$9, 2	/*Compara a entrada "\$1" com uma possibilidade de "A ganhar no "\$9"
	Se \$9 = \$1, segue o código, caso o contrário pula 2 linhas*/
ADDi \$E, \$E, 1	//\$E = \$E + 1 - Soma no placar de "A" que está no registrador \$E;
J 26	//Pula para a linha 26;
BNE \$1, \$A, 2	/*Compara a entrada "\$1" com uma possibilidade de "A ganhar no "\$A"
	Se \$A = \$1, segue o código, caso o contrário pula 2 linhas*/
ADDi \$E, \$E, 1	//\$E = \$E + 1 - Soma no placar de "A" que está no registrador \$E;
J 26	//Pula para a linha 26;
BNE \$1, \$B, 2	/*Compara a entrada "\$1" com uma possibilidade de "A ganhar no "\$A"
	Se \$A = \$1, segue o código, caso o contrário pula 2 linhas*/
ADDi \$E, \$E, 1	//\$E = \$E + 1 - Soma no placar de "A" que está no registrador \$E;
LW \$1, FF(\$0)	//(linha 26)Lê novamente a entrada paralela e joga para o registrador \$1;
SLTi \$C, \$1, 80	//Se \$1 < 80'h \$C = 1, \$C = 0 c.c.
BEQ \$0, \$C, FD	//Se \$C = 0, sobe duas linhas, caso o contrário ele continua o código;
J 0	//Pula para a linha 0;

- Código Assembly sem comentários:

```
LW $1, FF($0)
SLTi $2, $1, 80
BNE $0, $2, FD
ADDi $3, $0, 89
ADDi $4, $0, 92
ADDi $5, $0, A4
ADDi $6, $0, A1
ADDi $7, $0, 8A
ADDi $8, $0, 94
ADDi $9, $0, 91
ADDi $A, $0, A2
ADDi $B, $0, 8C
BNE $1, $3, 2
ADDi $F, $F, 1
J 26
BNE $1, $4, 2
ADDi $F, $F, 1
J 26
BNE $1, $5, 2
ADDi $F, $F, 1
J 26
BNE $1, $6, 2
ADDi $D, $D, 1
J 26
BNE $1, $7, 2
ADDi $D, $D, 1
J 26
BNE $1, $8, 2
ADDi $D, $D, 1
J 26
BNE $1, $9, 2
ADDi $E, $E, 1
J 26
BNE $1, $A, 2
ADDi $E, $E, 1
J 26
BNE $1, $B, 2
ADDi $E, $E, 1
LW $1, FF($0)
SLTi $C, $1, 80
BEQ $0, $C, FD
J 0
```