

MI SISTEMAS DIGITAIS 2014.2
TUTOR: Anfranserai Moraes Dias
DATA: 16/10/2014

COORDENADOR: Victor Figueiredo
QUADRO: Lucas Moraes
MESA: Diego Leite

RELATÓRIO DE MESA

IDEIAS:

- Criar um grupo de registradores para os OPCODES visando diminuir o endereço dos registradores.
- CALL -> 4 REGISTRADORES PARA PARÂMETROS
-> 2 REGISTRADORES PARA RETORNO
- Para as demais instruções que irão utilizar a instrução de soma da ula, a unidade irá enviar um function para a ula.
- Haverá um diagrama de classes para cada grupo funcional (Aritmética, load/store, hump, jpc)
- Diagrama de classes seriam quantizadas por módulo

FATOS:

- RET -> Procura o endereço do retorno na pilha
- O CALL salva o endereço atual do PC
- Como saber se há necessidade de ter um registrador no circuito:
 - 1 - Se quem tá providenciando o dado é volátil?
 - 2 - Se você tem alguém conectado ao barramento, se o barramento está sendo dividido por alguém?
- O RF do BRFL é o registrador R com o endereço para o pulo. O RF é na verdade R.
- O registrador de flags é um registrados interno

QUESTÕES:

- O endereço salvo no CALL pro retorno é o endereço onde há a chamada do CALL ou o próximo endereço?
- Como o processador identifica o final de um programa?
- Como garantir que as instruções gastem 5 ciclos por instrução?
- Como a ula vai saber em qual registrador deve gravar?
- Qual a definição de classe no ambiente de hardware?

METAS:

- Colocar o function do datapath da ULA # Victor
- Refatorar o caso de uso da ULA (A ação estava chamando outra) # Victor
- Refatorar o documento de requisitos, na seção do BRFL (Operandos) # Jadson
- Implementar o banco de registrados e a ULA (em Verilog) # Wagner, Filipe e Jadson

- Fazer o diagrama de classe da ula # Victor
- Criar os casos de uso das instruções: JR, JPC, CALL, RET, LW, SW, NOP, HALT e BRFL # Lucas, Caio, Diego, Matheus e Kelvin
- Pegar o modelo do documento de arquitetura, adicionar em nosso repositório e refatorá-lo. # Filipe