

# Roteiro do Projeto 3 - Grupo 04

MC723 Laboratório de Projeto de Sistemas Computacionais

084197 - Vinicius Dias de Oliveira Gardelli

103659 - Nicholas Matuzita Mizoguchi

119098 - André Vitor Terron

119319 - Diego Silva de Carvalho

138311 - Fabio Yudi Murakami

repositório: [git@github.com:andreterron/unicamp-mc723-lab3-grupo4.git](https://github.com:andreterron/unicamp-mc723-lab3-grupo4.git)

## Conceito

1) Para iniciar o processo, o sistema de hardware seleciona dinamicamente um dos processadores como bootstrap (BSP) e os demais como application processors (AP's). O BSP executa os códigos de configuração e inicialização da BIOS e inicializa os AP's. Depois que todos os processadores foram inicializados, o BSP começa a executar o código de inicialização do sistema operacional.

Depois, cada processador do *bus* principal realiza uma inicialização individual de processador (conhecido também como *hardware reset*) e um *built-in self test (BIST)* que é um teste para verificar o bom funcionamento. Um *hardware reset* define para cada registrador do processador um estado padrão. Cada processador também invalida toda suas caches internas, TLB's (*translation lookaside buffers*) e BTB (*branch target buffer*).

2) O controle concorrência em hardware é feita através de um conjunto específico de instruções, chamadas de instruções atômicas. Essas instruções podem ser usadas para fins de sincronização, uma vez que têm garantia de não serem interrompidas, podendo então modificar informações na memória compartilhada, permitindo os processadores identificarem o estado das execuções. Algumas instruções como de incremento, *swap* de dados entre um registrador e um local na memória e a instrução de comparação e *swap* fazem parte do conjunto de instruções atômicas.

3) A escolha do melhor trecho de código a ser implementado em hardware depende do fator que se quer otimizar. Como se quer acelerar o desempenho do software, deve-se analisar qual parte do código utiliza mais tempo de execução e a partir disso criar o modulo de hardware com essa parte do código.

## Metodologia

### Configuração do ambiente

- Criar um *router* para rotear a linha de transporte dos processadores.
- Instanciar os 8 processadores MIPs na *main.cpp*, criando as memórias, buses, inicializando as memórias, e outras configurações necessárias para a execução.
- Teste do funcionamento dos cores concorrentes, e a conexão de dados entre eles.

### Desenvolvimento

- Definir software e criar a implementação para rodar no ambiente criado.
- Analisar código através de uma ferramenta de profiling e encontrar um trecho de código possível de ser modularizado e transformado em hardware para otimizar a execução.
- Implementar o offload para hardware e testar seu uso na implementação feita.