Davi Friggi, Gustavo Tapajós, Rafael Figueredo

Profa. Denise Stringhini

Arquitetura e Organização de Computadores

Relatório do Simulador de Dispositivo Embarcado/IoT

Nosso projeto é sobre um Semáforo Inteligente com Sensor de Fluxo de Veículos. A ideia é simular o funcionamento de um semáforo inteligente que funcione de maneira semelhante aos semáforos inteligentes já implementados, inclusive na cidade de São José dos Campos.

Funcionamento da lógica

Primeiramente, nosso semáforo funciona com base em um sensor de fluxo de carros que armazena os últimos 5 valores lidos para a realização do cálculo de tempo para o sinal aberto. O valor desse cálculo é enviado a um contador decrescente para visualização do tempo que falta para o semáforo fechar, em adição ao próprio semáforo com as 3 cores padrão (vermelho, amarelo e verde). Todas essas implementações visuais foram feitas utilizando o Bitmap Display dentro do próprio Mars.

Nesse sentido, utilizamos um vetor com 5 posições de memória para armazenar os valores lidos pelo sensor, juntamente com um índice circular, o qual, após a utilização das 5 posições do vetor, começa a preenchê-lo novamente pela primeira posição. Outro aspecto importante foi a utilização de uma média móvel para o cálculo do tempo, a qual, em decorrência de um vetor com

índice circular, foi feita de maneira simples, somando os 5 valores do sensor de dividindo pelo tamanho (5).

Nessa perspectiva, após o cálculo do tempo pela média móvel, o valor calculado é enviado para o contador decrescente e para o alterador do semáforo, que muda para "aberto" (verde) e simula o decremento do tempo recebendo um input de caracter Enter do teclado, para tornar o modelo semelhante à passagem dos segundos. Após isso, o semáforo muda para "atenção" (amarelo) e utiliza um contador decrescente com o número 5 para simular 5 segundos com o sinal amarelo. Por fim, o semáforo volta para "pare" (vermelho) e aguarda pela próxima leitura do sensor, para realizar novamente todo o processo.

Código MIPS

Sob uma primeira análise, foi implementado um menu interativo para visualização do simulador:

"Selecione o processo desejado:

1 - Seguir com o processo

2 - Mostrar histórico últimos 5 semáforos

3 - Mostrar média móvel atual

4 - Encerrar"

sendo possível, seguir com a leitura do sensor (seguir processo), mostrar o histórico das últimas 5 leituras, realizar o cálculo da média móvel para o vetor histórico atual, e encerrar o programa.

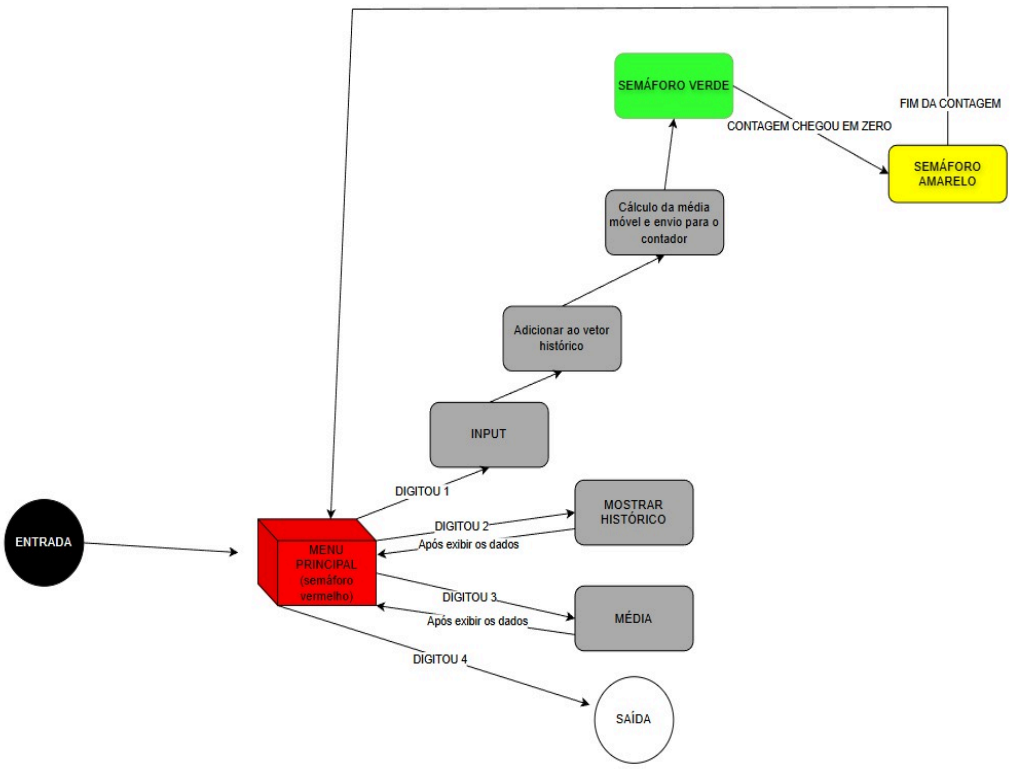
Para a implementação de cada um deles, foi necessário tanto o uso de opcodes do tipo branch quanto do tipo jump (jal e j), a depender se seria necessário armazenar o $ra em $sp. Nesse contexto, para as funções de adicionar ao histórico, calcular a média móvel e mudar a cor do semáforo, todas foram feitas com o jal e o empilhamento no stack pointer. Todas as variáveis de

parâmetros foram passadas utilizando os registradores do tipo $a, e as de retorno da função foram feitas utilizando registradores do tipo $v.

Ademais, o display Bitmap foi implementado utilizando endereços de memória interna do Mars. O primeiro endereço foi o 0x10008000. O Bitmap funciona com base no processo de colorir um pixel a cada vez, portanto o endereço ***0(0x10008000)*** acessa o primeiro pixel do display, o endereço ***4(0x10008000)*** acessa o segundo pixel, e assim por diante. Para a cores foram utilizadas variáveis .word com valores hexadecimais, por exemplo: *COR_VERMELHO: .word 0x00FF0000.*

Por fim, para o contador foram feitos "desenhos" dentro do Bitmap para simular o decremento do valor. Foi necessário pintar 20 números diferentes, 10, de 0 a 9, para o primeiro dígito, e 10, de 0 a 9, para o segundo diígito. Nesse sentido, por meio da divisão por 10, conseguimos separar o dígito das unidades (utilizando o resto da divisão, *mfhi*) do dígitos das desenhos (utilizando o quociente, *mflo*), cada um sendo passado para uma função que reiniciava o display a cada decremento e "pintava" novamente o novo número.

Diagrama de Estados



Uso de Inteligência Artificial

Sugestão de Indíce Circular para o vetor de histórico de carros	Método utilizado para considerarmos somente as últimas 5 leituras do vetor histórico de carros, de maneira que fosse substituida a leitura mais "antiga"
Sugestão do cálculo para o ajuste de tempo do semáforo	Cálculo sugerido foi o de somar todos os 5 valores do vetor histórico e dividir por 5, utilizando o algoritmo não trivial da média móvel.
Verificação de funcionamento dos primeiros passo com a IA	Essa verificação ocorria pelo fato de o display e a parte visual não estarem prontas. O foco era o funcionamento lógico, portanto, essas aferições eram feitas usando a IA.

Sugestão do Shift Left Logical para calcular a posição * 4	Sugestão de trocar um mult com 4 por um sll de 2 bits, para fazer a multiplicação do índice posição * 4, garantindo o correto acesso ao vetor.

O que corrigimos da IA

Utilizamos o Gemini, e ele havia “corrigido” a função adiciona_carro_historico e adicionou um processo desnecessário de salvar o $ra em $sp. Esse processo é desnecessário por conta de não haver outra chamada de função para que o $ra fosse sobrescrito, portanto essa correção salvou espaço no stack pointer.