

Grupo: **Felipe Kenzo Araki – 2022005633**
Glauber da Silva Moura – 2022000299
Isabelle Francine Guedes Romão – 2021029710
Kauan Barbosa da Silva - 2022010132
Pedro Andrade Gomes - 2022006926

Relatório II

Conversão analógico para digital de $m(t)$ com o MCP3002 colocando os bits sequencialmente na FPGA

O sistema em questão realiza a aquisição de dados em um fluxo serial de bits, processando-os em amostras de 8 bits, descartando os 2 bits menos significativos de uma amostra original de 10 bits para manter uma resolução adequada. O processo é controlado por sinais de clock em diferentes frequências, garantindo a sincronização e o tempo de amostragem necessário.

No fluxo de amostragem temos:

Entrada: Um fluxo serial de 10 bits por amostra (proveniente de um conversor analógico-digital, como o MCP3002).

Saída: Uma amostra de 8 bits, obtida descartando os 2 bits menos significativos para evitar perda excessiva de resolução.

A cada pulso de clock, 1 bit é deslocado para a saída. Após 8 pulsos de clock, uma amostra completa de 8 bits é formada. Como a entrada original tem 10 bits, os últimos 2 bits são ignorados para simplificação do processamento.

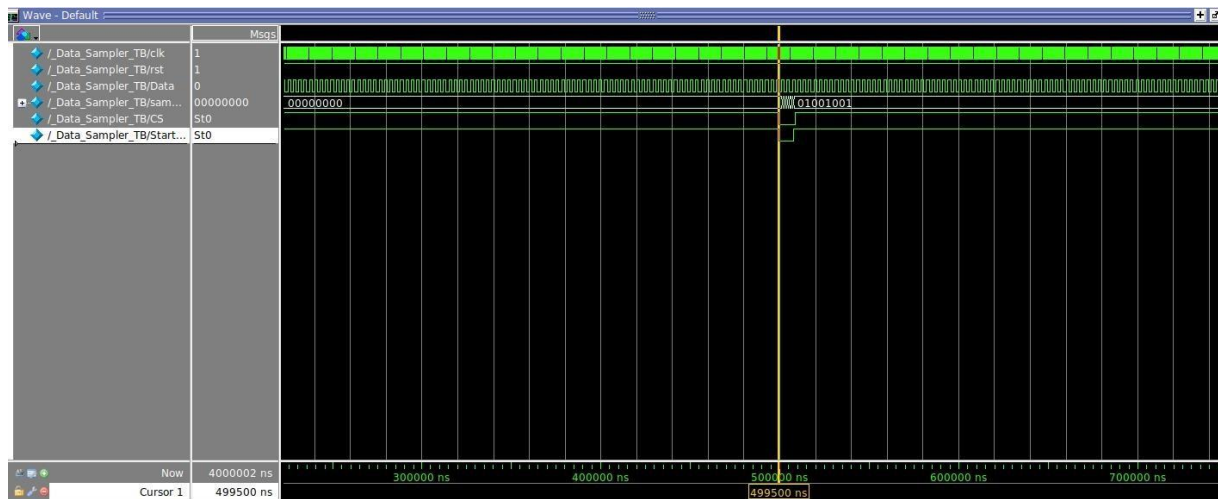
Em relação ao tempo, temos:

A cada 5 ms, uma amostra completa (8 bits) deve ser capturada. Isso resulta em uma taxa de amostragem de 200 Hz (1 amostra a cada 5 ms).

- Clock Base (Principal): 50 MHz (usado para sincronização geral da placa).
- Clock do Conversor (MCP): 50 kHz (frequência de alimentação do MCP3008).
- Clock Interno (Processamento): 200 Hz (utilizado para controle interno da placa, como conversão e processamento de dados).

O chip (MCP3002) é ativado no início do processo de amostragem. Após a captura dos 8 bits necessários, o chip é desligado para economizar energia e evitar leituras desnecessárias. Durante o processo, o sistema opera com 3 clocks distintos:

- Clock Base (50 MHz): Sincronização principal da placa.
- Clock do MCP (50 kHz): Controla a comunicação com o conversor A/D.
- Clock Interno (200 Hz): Gerencia o processamento interno (filtragem, conversão, etc.).



Abaixo temos o código utilizado para a simulação:

```

module _Data_Sampler
(
    input clk,          //deve ser alimentado com 50khz
    input rst,
    input Data,

    output reg CS,
    output reg StartConv,          //flag de controle de conversao
    output reg [7:0] sampler      //registro de dados
);

    reg [3:0] countSamp;          //contador de controle de conversao
    reg [7:0] countCtrl;          //contador de controle de frequencia de taxa de amostragem = 200Hz

    //always de controle: contador
    always@(posedge clk or negedge rst)
        begin
            if(~rst)
                begin
                    countCtrl <= 0;
                    StartConv <= 1; //reset em nivel alto
                end else
                begin
                    //se bate o tempo do periodo de 5ms, reseta o contador e habilita a conversao
                    if(countCtrl == 249)
                        begin
                            countCtrl <= 0;
                            StartConv <= 0;
                        end else
                        begin
                            //quando a conversao terminar, 10 ciclos de clock, desabilita a conversao
                            if(countCtrl == 7)
                                begin
                                    StartConv <= 1;
                                    countCtrl <= countCtrl + 1;
                                end
                            else
                                countCtrl <= countCtrl + 1;
                        end
                    else
                        countCtrl <= countCtrl + 1;
                end
            else
                countCtrl <= countCtrl + 1;
        end

```

```

//incrementa
    else
countCtrl <= countCtrl + 1;
    end
    end
    end

//o mcp atualiza na borda de descida
always@(posedge clk or negedge StartConv or negedge rst)
    begin
        if(~rst)
            begin
                CS <= 1; //coloca CS em nível alto=desabilita conversor
                sampler <= 0; //zera o registro de controle
                countSamp <= 0; //zera o contador de controle
            end else
            begin
                if(~StartConv)
                    begin
                        if(countSamp == 0)
                            begin
                                CS <= 0;
                                countSamp <= countSamp + 1;
                            end
                        else
                            begin
                                CS <= 0;
                                countSamp <= countSamp + 1;
                                sampler <= (sampler << 1) | Data;
                            end
                        end
                    end
                else
                    //se StartConv for desabilitado, zera o contador e desabilita CS
                    begin
                        countSamp <= 0;
                        CS <= 1;
                        //sampler <= 0;
                        //nao preciso zerar o contador, a interacao natural desse always
                        //com o outro fara isso naturalmente
                        //depois disso, sampler contera o valor dos 8 bits mais significativos do
                        //MCP3002 referente ao sinal convertido
                    end
                end
            end
        endmodule

```

O sistema realiza uma amostragem eficiente de dados em um fluxo serial, convertendo amostras de 10 bits em 8 bits a cada 5 ms. A utilização de múltiplos clocks (50 MHz, 50 kHz e 200 Hz) permite um controle preciso do tempo de amostragem e processamento interno, enquanto a desativação do chip após a captura garante economia de energia. A remoção dos 2 bits menos significativos mantém uma resolução adequada para a aplicação proposta.