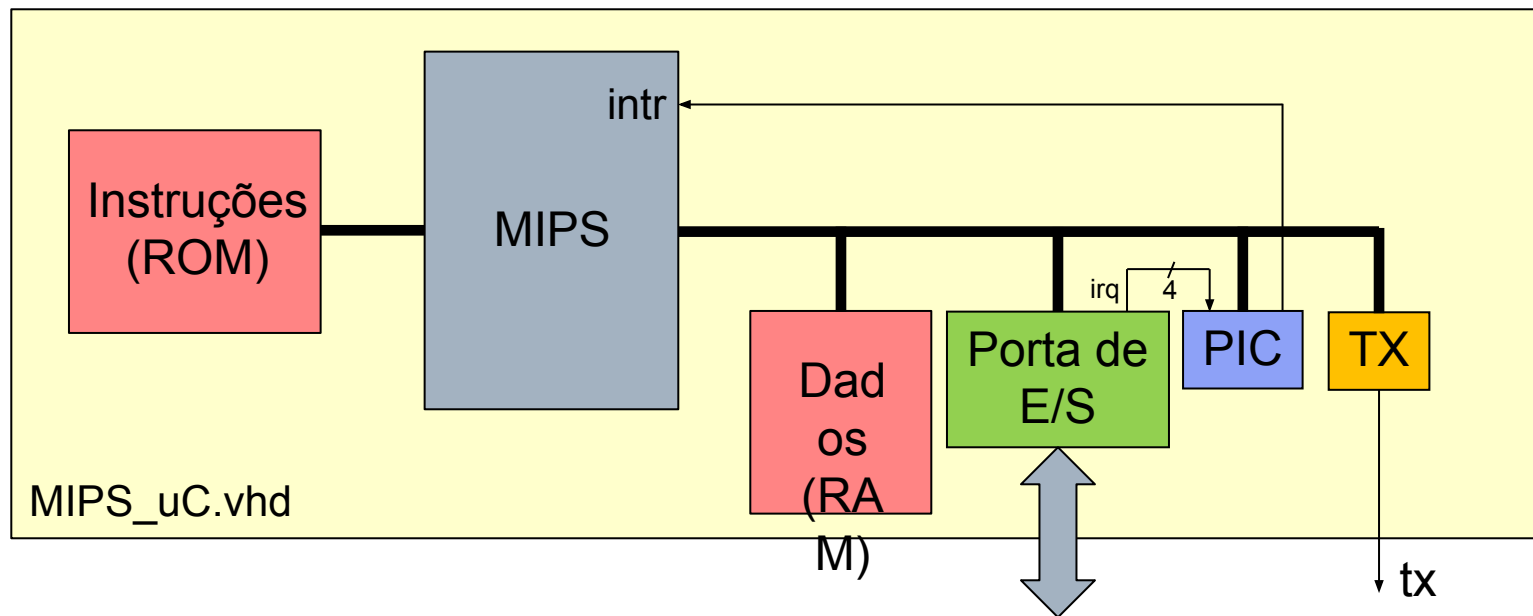


Trabalho 5 - parte 1

- Adicionar módulo de transmissão serial ao MIPS_uC
 - Na interface do MIPS_uC deve ser adicionada uma saída serial de 1 bit (*tx*)
 - Módulo de transmissão/recepção serial no *moodle*



Trabalho 5 - parte 1

□ Módulo transmissor serial

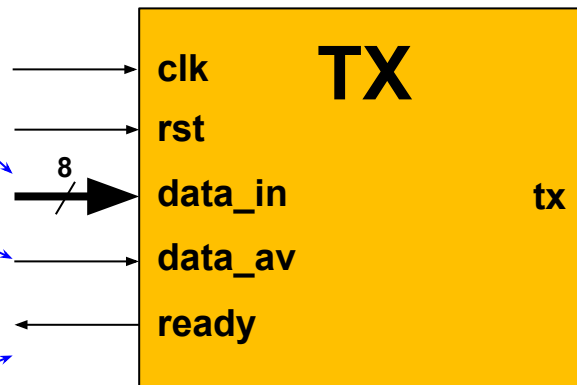
- Transmite 8 bits de dados seguindo o padrão RS232

1. Start bit (0)
2. Data (8 bits)
3. Stop bit (1)

Dado a ser serializado

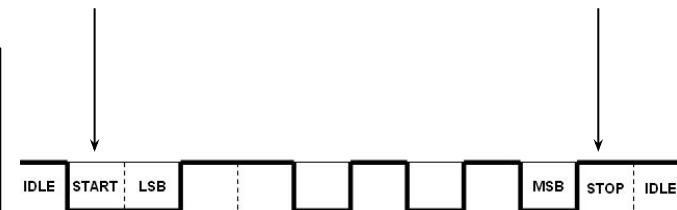
Indica dado válido em *data_in* (deve ficar ativo por 1 ciclo de *clock*)

Indica que o módulo está apto a transmitir (1)



Start bit (0)

Stop bit (1)



Data: 01010110

Trabalho 5 - parte 1

□ Módulo transmissor serial

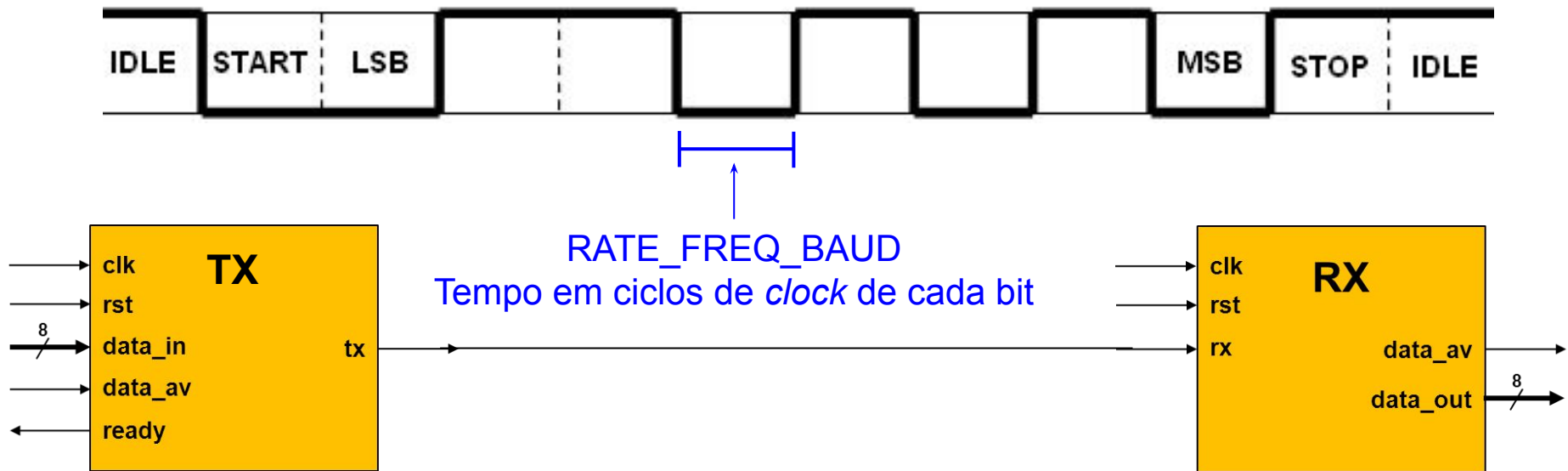
- O parâmetro RATE_FREQ_BAUD deve ser setado com a razão entre a frequência de operação do módulo em (Hz) e a velocidade de transmissão
- Exemplo: Módulo operando a 50MHz e transmissão de 115200 bps
 - $\text{RATE_FREQ_BAUD} = 50000000/115200$

```
entity UART_TX is
    generic(
        RATE_FREQ_BAUD : integer
    );
    port(
        clk             : in std_logic;
        rst             : in std_logic;
        ...
```

Trabalho 5 - parte 1

□ Módulo transmissor serial

- O parâmetro RATE_FREQ_BAUD deve ser setado com a razão entre a frequência de operação do módulo em (Hz) e a velocidade de transmissão



Trabalho 5 - parte 1

- Implementar em *assembly* as seguintes funções
 - *void PrintString (char *string)*
 - Manda todos os caracteres de uma *string* para o módulo de transmissão serial
 - Considerar que o final de uma *string* é indicado pelo byte 0
 - *char *IntegerToString (int n)*
 - A função deve converter o número *n* em uma *string* finalizada com o byte 0
 - Retorna um ponteiro para a área de memória onde está a *string* correspondente ao parâmetro *n*
-

Trabalho 5 - parte 1

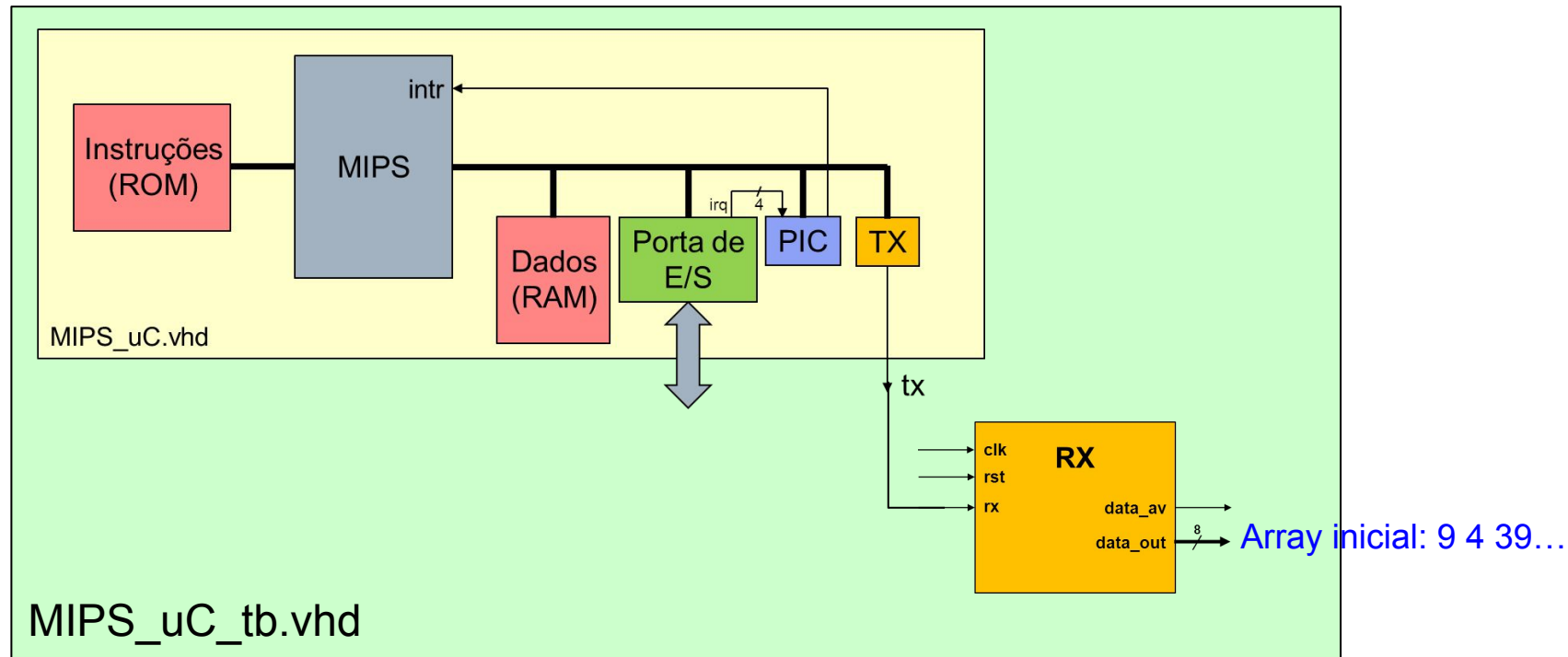
■ Aplicação

- Mesma do trabalho 4 – parte 2 (*Bubble sort* + *Cryptos*)
 - As funções implementadas devem ser utilizadas pelo *BubbleSort* e pelos *Handlers* para enviar ao módulo de transmissão serial
 - *BubbleSort*
 - “Array inicial: 9 4 39 ...”
 - “Array final: 1 2 3 ...”
 - *Handlers*
 - Mensagens dos *CriptoMessages* (não precisam ser armazenadas)
-

Trabalho 5 - parte 1

■ *Test bench*

- ❑ Ligar o módulo de recepção serial (RX) no MIPS_uC
- ❑ Olhar a saída de dados do módulo RX como ASCII



Trabalho 5 - parte 1

□ Estrutura do código na memória

