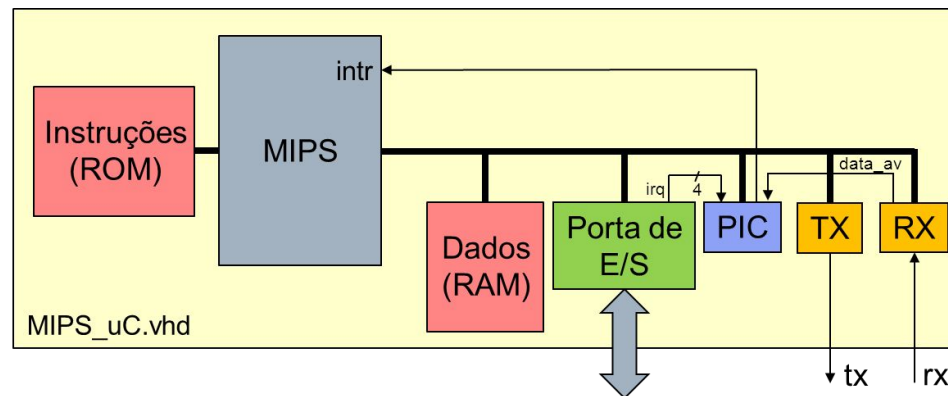


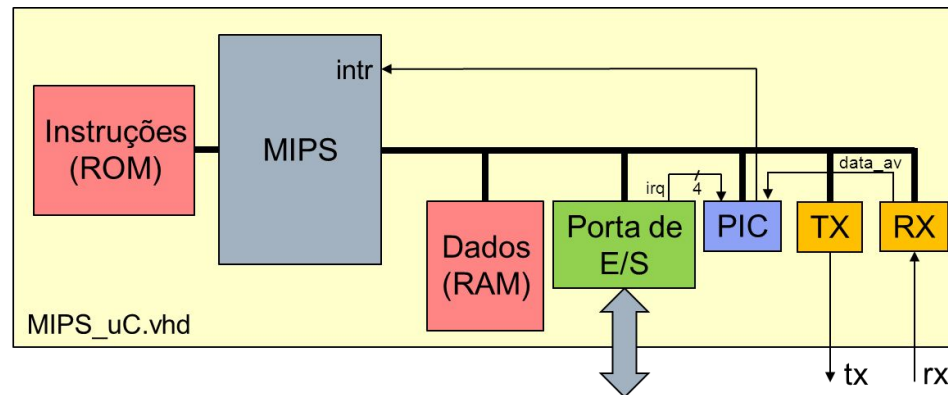
Trabalho 6 – parte 1

- Adicionar módulo de recepção serial ao MIPS_uC
 - Na interface do MIPS_uC deve ser adicionada a entrada serial de 1 bit (*rx*)
 - A saída *data_av* do módulo RX deve ser ligada à *irq(3)* do PIC a fim de interromper o processador sempre um dado está disponível na saída *data_out*
 - Adicionar registrador *RATE_FREQ_BAUD* nos módulos TX e RX a fim de programar velocidade via software (alterar a interface dos módulos)



Trabalho 6 – parte 1

- Modificar o MIPS_uC de maneira que seja possível carregar as memórias com programa/dados através do módulo RX
 - Os grupos tem total liberdade para modificar o MIPS_uC
 - O programa/dados devem ser transmitidos utilizando um programa de comunicação serial (e.g. Hyper Terminal)
 - As imagens das memórias devem ser binárias (MARS)
 - Apresentar diagrama detalhado do MIPS_uC a fim de suportar a carga das memórias



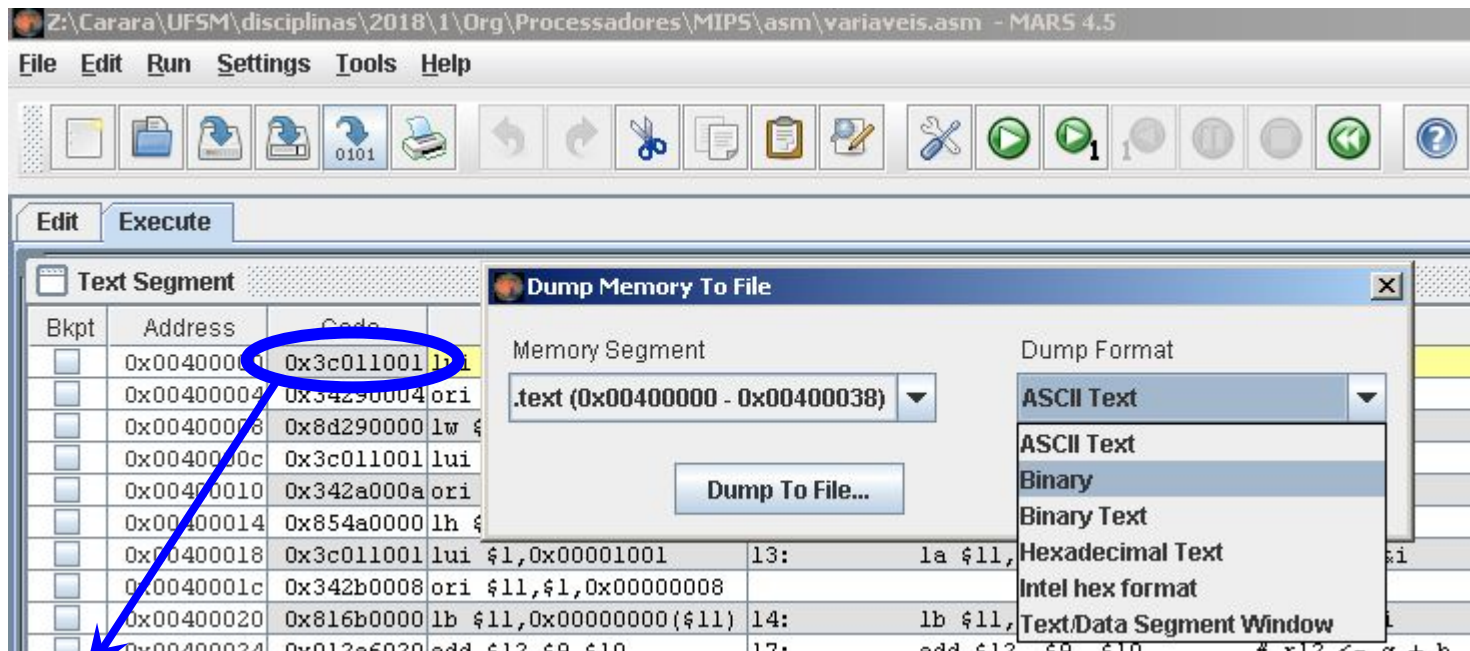
Trabalho 6 – parte 1

☐ Sugestão

- Utilizar os *slide-switches* da placa para selecionar o modo de operação do MIPS_uC
 - ☐ Modo de programação da memória de instruções
 - ☐ Modo de programação da memória de dados
 - ☐ Modo de execução
- Adicionar à interface do MIPS_uC entradas correspondente aos *slide-switches* que forem utilizados

Trabalho 6 – parte 1

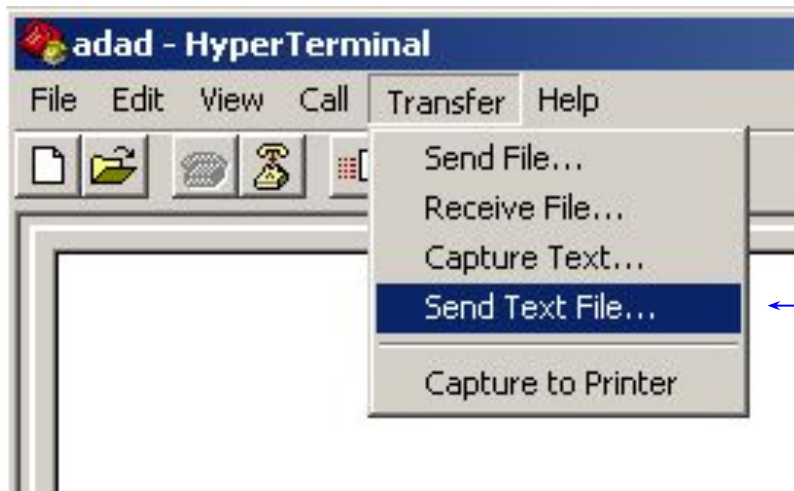
- Gerando imagens binárias com o MARS
 - Atenção ao ordenamento dos bytes no arquivo binário (*little-endian*)



01 10 01 3c 04 00 29 34 00 00 29 8d 01 10 01 3c 0a 00 2a 34
00 00 4a 85 01 10 01 3c 08 00 2b 34 00 00 6b 81 ...

Trabalho 6 – parte 1

- Enviando arquivos utilizando o *Hyper Terminal*
 - Qualquer outro programa de comunicação serial pode ser utilizado, desde que seja possível enviar um arquivo



← Selecionar o arquivo .bin

Trabalho 6 – parte 1

- Enviando arquivos utilizando a linha de comando no windows
 - Utiliza-se os comandos
 - *mode*: configurar a porta serial
 - *copy*: enviar o arquivo pela porta serial
 - Exemplo
 - *mode* COM3 BAUD=115200 PARITY=n DATA=8
 - *copy* arquivo.bin COM3
-

Trabalho 6 – parte 1

☐ Aplicações

- As aplicações devem ser carregadas via módulo RX
 - ☐ O programa principal deve configurar a velocidade de comunicação dos módulos para 57600 bps e entrar em um *loop* infinito esperando pela carga de uma aplicação
 - App1: contador
 - ☐ Utilizar o contador com incremento de 1 segundo dos trabalhos anteriores
 - ☐ O código deve utilizar variáveis
 - App2: Echo (semelhante ao `uart_terminal.bit`)
 - ☐ Adicionar um *handler* para tratar dados recebidos pelo módulo RX através de um programa de comunicação serial (e.g. PuttY) quando uma tecla é pressionada
 - ☐ O *handler* deve simplesmente enviar de volta para o terminal (via módulo TX) o código ASCII da tecla pressionada
-

