

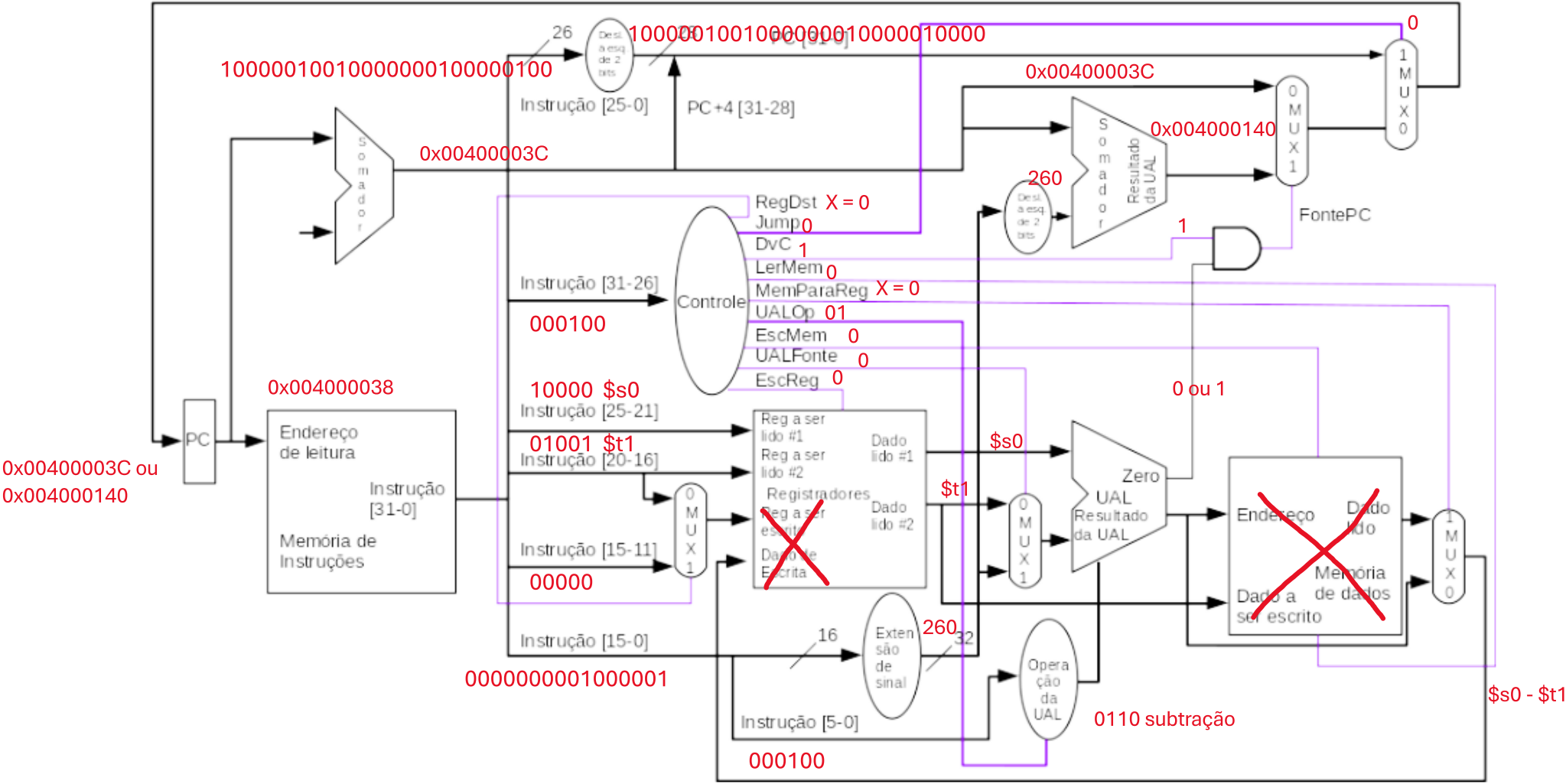
conversão para linguagem de máquina

\$s0 = 16; \$t1 = 9; beq = 4

imm = 0000 0000 0100 0001

$imm = (0x00400140 - 0x004000038 + 4) / 4$
 $imm = (0x00400140 - 0x00400003C) / 4$
 $imm = 0x000000104 / 4$
 $imm = 0x000000041$

000100 10000 01001 0000000001000001
opcode rs rt imm



Na borda de subida do sinal de relógio o processador recebe o endereço da próxima instrução, neste caso, a instrução `beq $s0, $t1`, loop no endereço $0x004000038$.

Dada a instrução, primeiro deve-se decodificá-la, separando os campos dela. Também é adicionado 4 a PC em um somador para carregar o endereço da próxima instrução e é feito um shift para a esquerda nos bits da instrução [25-0] e são concatenados com os 4 bits de PC+4, obtendo assim o endereço para o pulo caso a instrução fosse uma do tipo J.

Os campos da instrução atual são separados, e enviados para diferentes partes do processador. O opcode(instrução [31-26]) é decodificado pelo controle e são gerados os sinais de controle. $Jump = 0$; $DvC = 1$; $LerMem = 0$; $UALOp = 1$; $EscMem = 0$; $UALFonte = 0$; $EscReg = 0$.

É feita a extensão do sinal do imediato que passa de 16 para 32 bits, antes do imediato chegar no somador é feito um shift para a esquerda, que equivale a uma multiplicação por 4, que será somada a PC+4, chegando assim no endereço do pulo.

Pelo valor de $UALOp$ é carregada a operação que será realizada na UAL, com o sinal 01 é carregada a operação 0110, de subtração. A subtração feita na UAL é dos registradores $\$s0$ e $\$t1$, caso o resultado seja 0 ($\$s0 = \$t1$), a saída Zero da UAL vai mandar o valor 1 para FontePC o que quer dizer que a condição de desvio é verdadeira e a próxima instrução carregada no processador seria localizada pelo imediato, caso a saída de Zero for um 0 ($\$s0 \neq \$t1$), a condição para o pulo seria falsa e a próxima instrução carregada no processador seria PC+4.