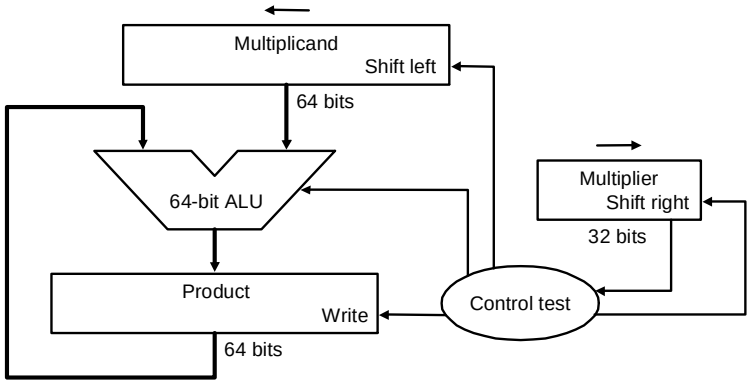
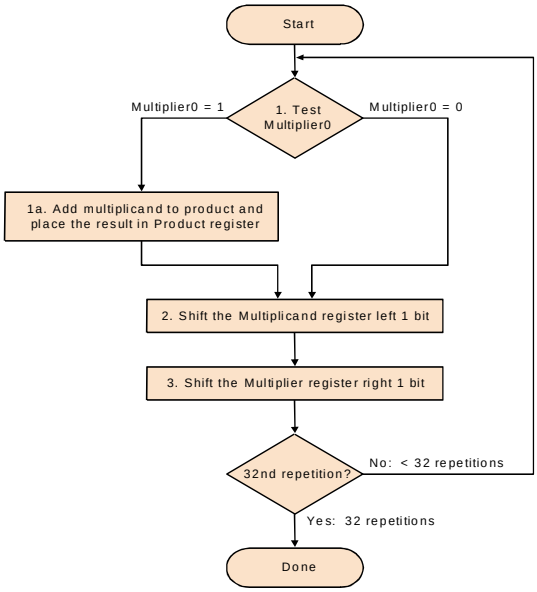
Exercício MC 404 - 26/08

Multiplicar dois números dados (armazenados em dois registradores no começo do código), acumulando o resultado em X, e no final do programa, armazenar o resultado em big endian no começo da memória RAM, usando Y.

OBS.: O código descrito, coloca a instrução 'mul fat1, fat2l' apenas para conferir o resultado (ela armazena-o em little endian em R0 e R1).

Hardware Multiplicador – Primeira versão






```
.nolist
.include "m88def.inc"
.list

Definição de nomes de variáveis a serem usadas no programa.
.def fat1=r16     ;.
def fat2l=r17
.def fat2h=r18

.org 0
rjmp main

main:

; desenvolver o programa
```

```
.nolist
.include "m88def.inc"
.list

.def fat1=r16     ; Definição de nomes de variáveis a serem usadas no programa.
.def fat2l=r17
.def fat2h=r18
.org 0
rjmp main

main:
    ldi xl, 0                    ; Inicializa palavra X, que armazenará o resultado.
    ldi xh, 0
    ldi yl, low(sram_start)    ; Palavra Y guardará o endereço de início da RAM.
    ldi yh, high(sram_start)
    ldi fat1, 100     ; Carrega primeiro e segundo fatores da multiplicação, com 8
    bits cada.
    ldi fat2l, 41
    ldi fat2h, 0   ; O segundo fator é armazenado em 16 bits, mas tem  8 bits.
    mul fat1, fat2l  ; Teste.
```

```
loop:
    bst fat1, 0       ; Verifica se o dígito atual do 1º fator é 1.
    brtc shift
    add xl, fat2l     ; Se for, soma 2º fator deslocado no acumulador do produto.
    adc xh, fat2h

shift:
    lsr fat1          ; Atualiza bit que será verificado do 1º fator.
    lsl fat2l         ; Ajusta deslocamento do 2º fator para próxima iteração.
    rol fat2h

    cpi fat1, 0       ; Verifica se ainda há bits diferentes de 0 a serem analisado
    no 1º fator.
    brne loop

    st y+, xh         ; Armazena resultado da multiplicação no início da memória
    RAM.
    st y, xl
fim:
    rjmp fim
```