FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA
Departamento de Engenharia Electrotécnica

Trabalho Final

Microprocessadores

Turno P2

19/12/2017

Alunos:

Henrique Joaquim 51006

Serafim Ciobanu 50006

Índice

1. INTRODUÇÃO

O objetivo do trabalho é a implementação de uma folha de cálculo básica onde seja possível realizar operações aritméticas, soma, subtração, multiplicação e divisão; importar folhas previamente existentes; exportar folhas em utilização bem como guardar o progresso sem que seja preciso fazer “export”.

Neste relatório abordar-se-ão tópicos referentes a este sistema, tais como:

- Descrição das opções, que consiste na descrição das várias opções das quais o utilizador poderá escolher;
- Abordar a grande maioria das funções escritas bem como explicar o seu funcionamento e a sua utilidade no contexto; para isso as mesmas vão ser descritas e, no caso das funções mais complexas ou de maior relevo apresentado o respetivo fluxograma para uma melhor perceção destas.
- Conclusões, que refletem sobre o cumprimento dos objetivos e requisitos iniciais, bem como problemas identificados durante o desenvolvimento do projeto;
- Referências bibliográficas, em que serão abordados assuntos detalhados dos temas tratados durante o relatório, retirados de fontes externas;

2. DESCRIÇÃO DA FASE INICIAL DO CÓDIGO

A primeira coisa a ser "performed" é verificar a existência do ficheiro "contents.bin", caso o mesmo não exista, é necessário criá-lo; se existir saltamos esse passo.

Se existir, lê-se o ficheiro e colocam-se os valores na fórmula e na matriz (por ser um ficheiro binário não é preciso grande preocupação com a escrita/leitura deste ficheiro).

Notar que: As variáveis "form_buffer" e "matrix" foram inicializadas " de seguida ", deste modo basta-nos passar um único endereço para realizar a leitura do ficheiro e um único número de bytes (21 : 16 matriz + 5 formula)

Quer exista quer não, após ser lido ou ter sido criado vamos para o contexto do programa:

É feito:

-Set_Video Mode

-ShowMouse

-ShowCursor

Após isto é chamada a função "menu" que será explicada adiante.

3. FUNÇÕES DO MENU

MENU FUNCTION

Nesta função é detetado o "clique" do utilizador e chama-se a função escolhida pelo mesmo.

Só é possível sair efetivamente do MENU quando se entra na função EXIT.

IMPORT SPREADSHEED

Nesta opção o utilizador pode importar uma folha de cáculo previamente existente;

Para isso bastar-lhe-á introduzir o nome do ficheiro.

Descrição da função:

-Antes de começar a ler o ficheiro, procedemos a uma "limpeza" do buffer de dados, de forma a evitar conflitos com valores previamente existentes; assim temos acerteza que só fica na folha de cálculo aquilo que de facto o utilizador quer.

-Após isto é pedido ao utilizador que introduza o nome do ficheiro. (Notar que: o utilizador deve seguir o exemplo dado: exemplo.txt, caso nao introduza .txt no final do nome do ficheiro irá dar erro e retornar ao menu).

-Depois destes dois passos passamos efetivamente a abrir o ficheiro e a lê-lo, este processo é feito através das funções File_Open, File_Read e File_Close. (Notar que: é importante fechar o ficheiro, uma vez que já não precisamos dele.)

-Agora o que está escrito no ficheiro está num buffer vamos proceder à "inspeção do mesmo". Este processo irá ser explicado através de um fluxograma.

Notar que:

-se toda a sintax do ficheiro estiver correta contudo o valor de uma células (ou mais) estiver corrompido, i.e. com valores excessivamente grandes ou valores não numéricos, a leitura da folha de cálculo não dá erro de imediato, não afetando assim todos os outros valores. Simplesmente na célula corrompida, é colocado o valor '-1'.

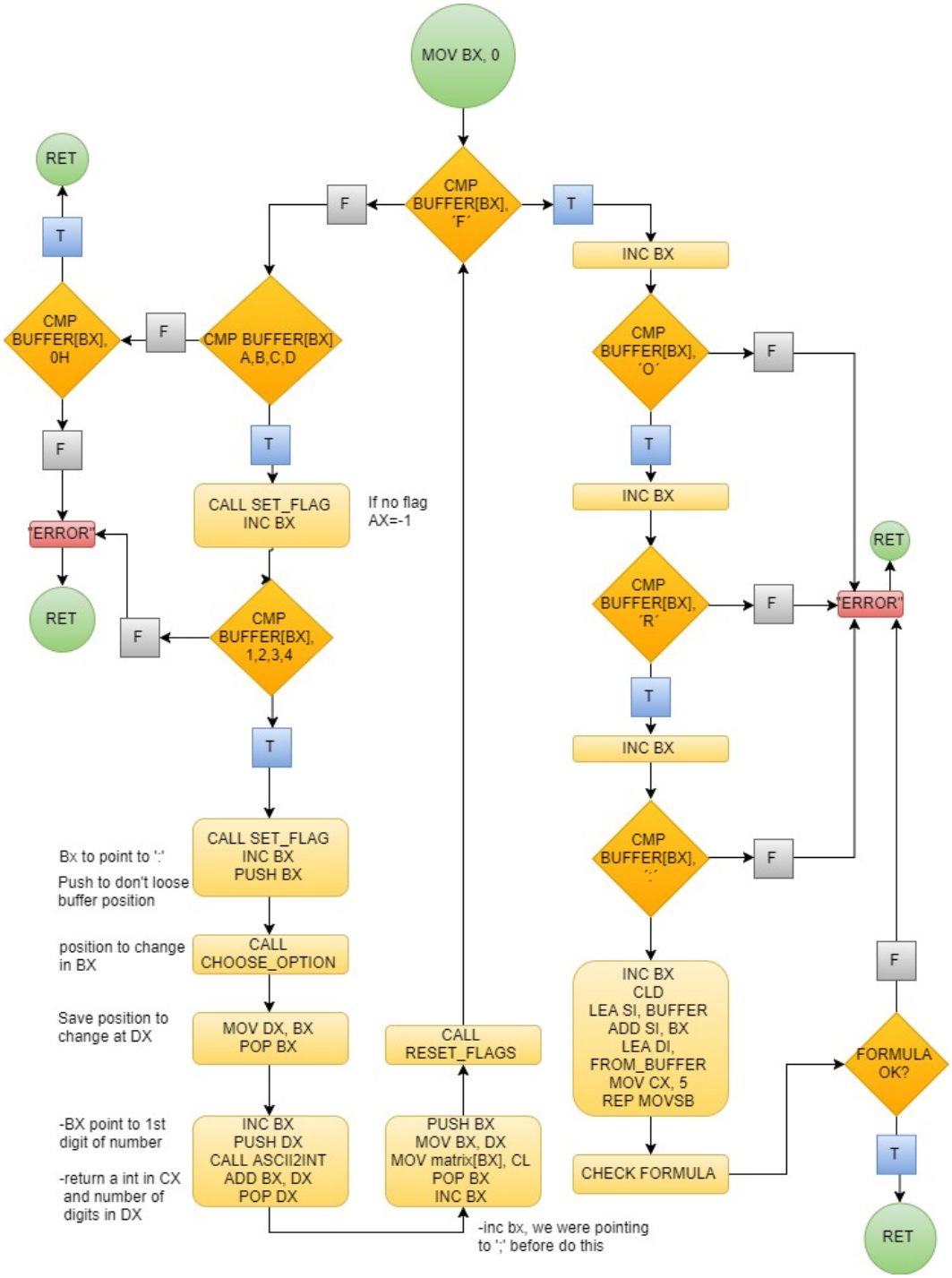
-se o "import" for mal sucedido restauramos os valores anteriormente na folha de cálculo. Para isso lêmos o ficheiro Contents.bin.

-no fim é impr

essa uma mensagem que informa o utilizador do sucesso/insucesso.

IMPORT SPREADSHEED

FLUXOGRAMA



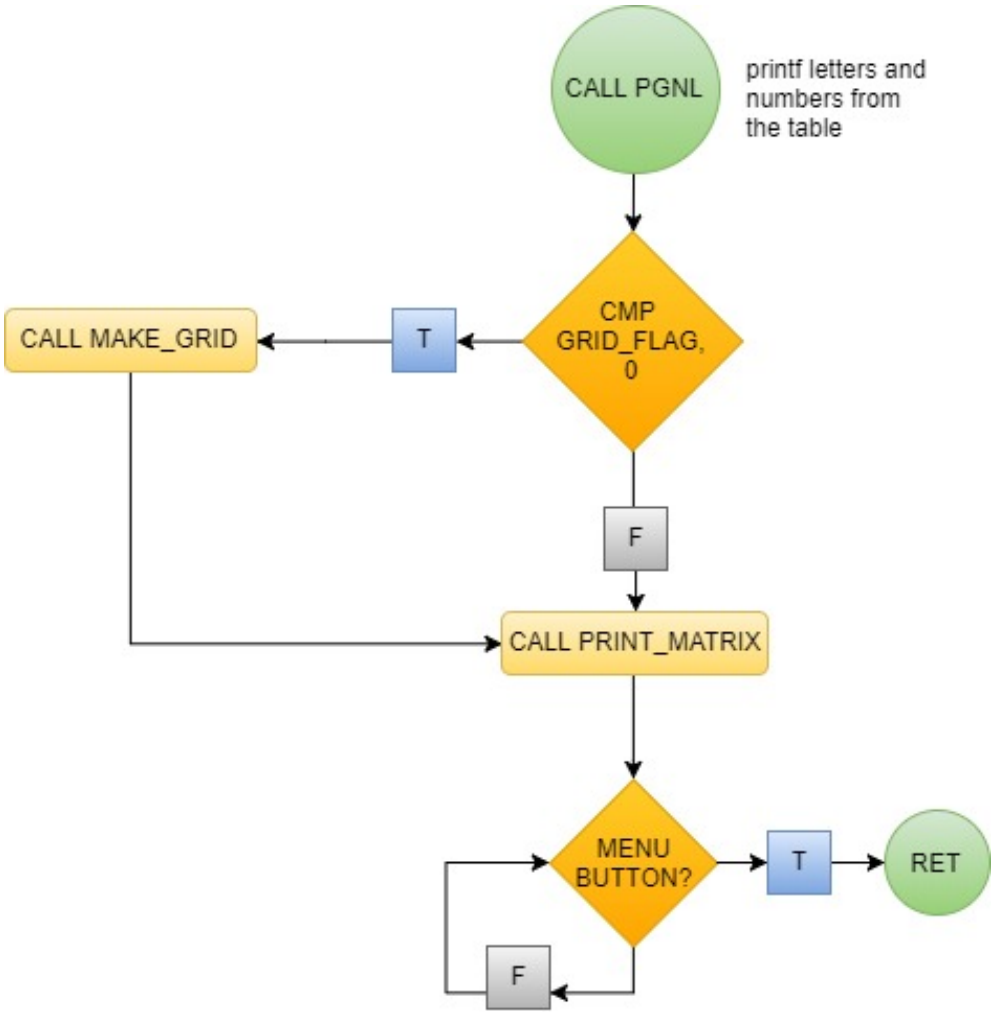
SHOW SPREAD

Esta função somente mostra os valores na tabela, com ou sem grid, depende da opção escolhida pelo utilizador previamente.

Após os valores serem impressos, a função aguarda que se clique no botão "MENU" para regressar ao menu.

SHOW SPREAD

FLUXOGRAMA



EDIT SPREADSHEET

Nesta função o utilizador pode alterar o valor das células e realizar operações entr as mesmas.

O primeiro passo é esperar o "clique" do utilizador; sendo que esse clique pode ser direcionado para:

-alterar uma célula

-alterar a fórmula

-regressar ao menu

Se for preciso alterar uma célula, consoante a posição clicada serão acionadas flags correspondentes à mesma (Notar que: antes de se realizar o próxima "clique" é dado um reset nas flags);

Após isto a função Choose_Option entra em ação para determinar a posição (0-15) da matriz que deve ser alterada; (lembrar que a função Choose_Option lê as flags e consoante as mesmas decide a posição a alterar, devolvendo-a em BX).

Depois de saber a posição a alterar o utilizador introduz o valor (SCAN NUM) (se ocorrer erro célula = -1) e este é guardado na matriz; após isto a célula é "limpa" e escrito o valor que efetivamente ficou guardado na matriz.

Após estes passos o resultado é atualizado (Res_Update).

Caso seja preciso alterar a fórmula chama-se a função Form_Reader.

EXPORT SPREADSHEET

A primeira coisa que fazemos nesta função é limpar o buffer, uma vez que não vamos querer guardar "lixo" no nosso ficheiro.

É pedido ao utilizador que introduza um nome para o ficheiro de acordo com o exemplo (exemple.txt), caso o utilizador introduza o nome sem ".txt" vai ocorrer erro!

Notar que não são guardados zeros; se não houver fórmula esta também não será guardada (nem "FOR:").

Para que se possa escrever no BUFFER_EXPORT somente as posições com conteúdo foi criado uma string "A1: A2: A3: (...) D4 FOR:" ; se a posição for de interesse vai buscar-se a este buffer a letra e o número respetivo e copia-se para o BUFFER_EXPORT.

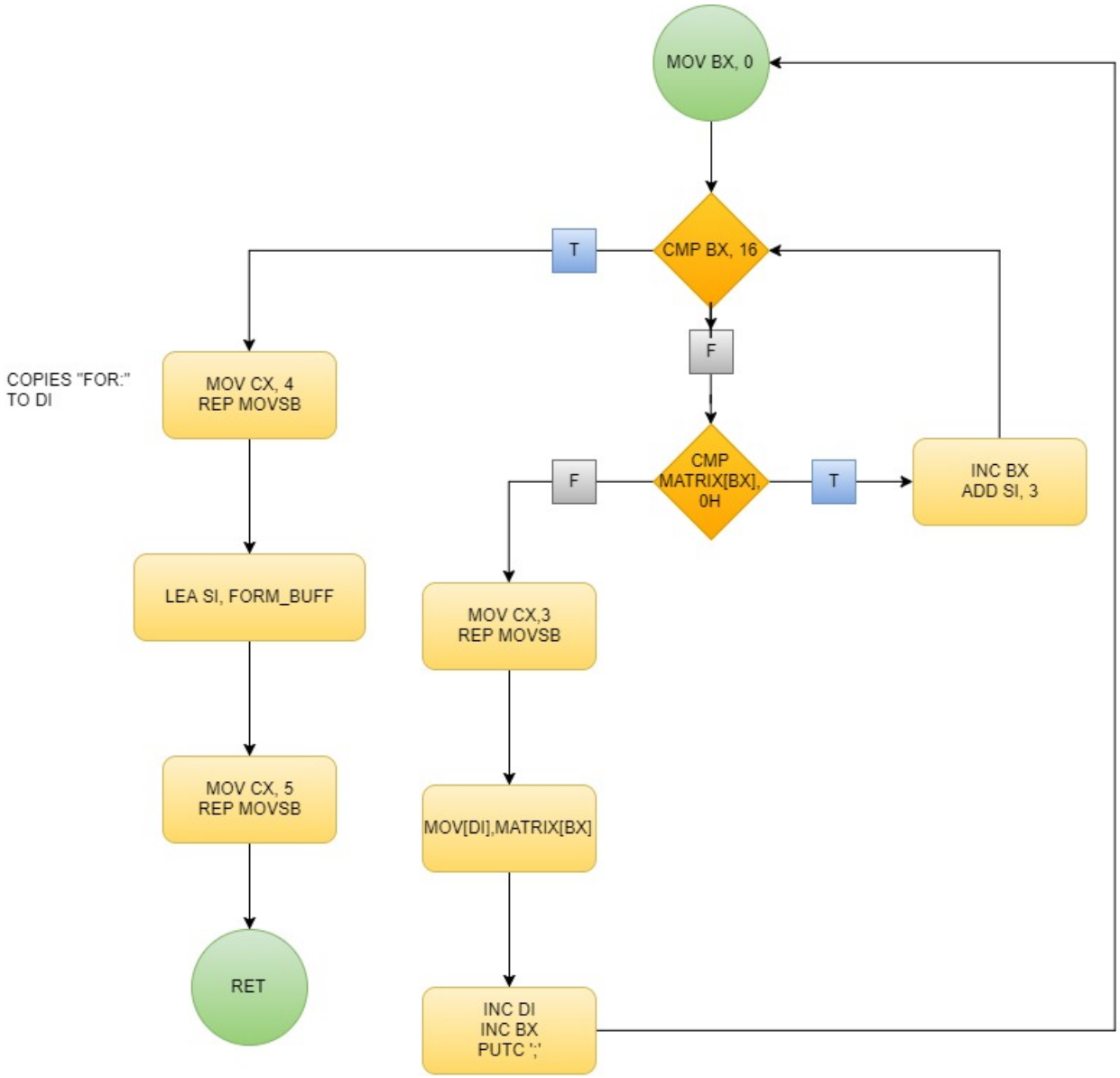
Após verificar as 16 posições da matriz e escrever as relevantes, é copiada a str "FOR:" e a respetiva fórmula parao EXPORT_BUFFER. (caso exista)

Depois de tudo o que se quer guardar estar no buffer vai proceder-se à escrita no ficheiro (Fopen --> Fwrite --> Fclose)

É possível uma melhor compreensão do processo "export" através do seu fluxograma.

EXPORT SPREADSHEET

FLUXOGRAMA

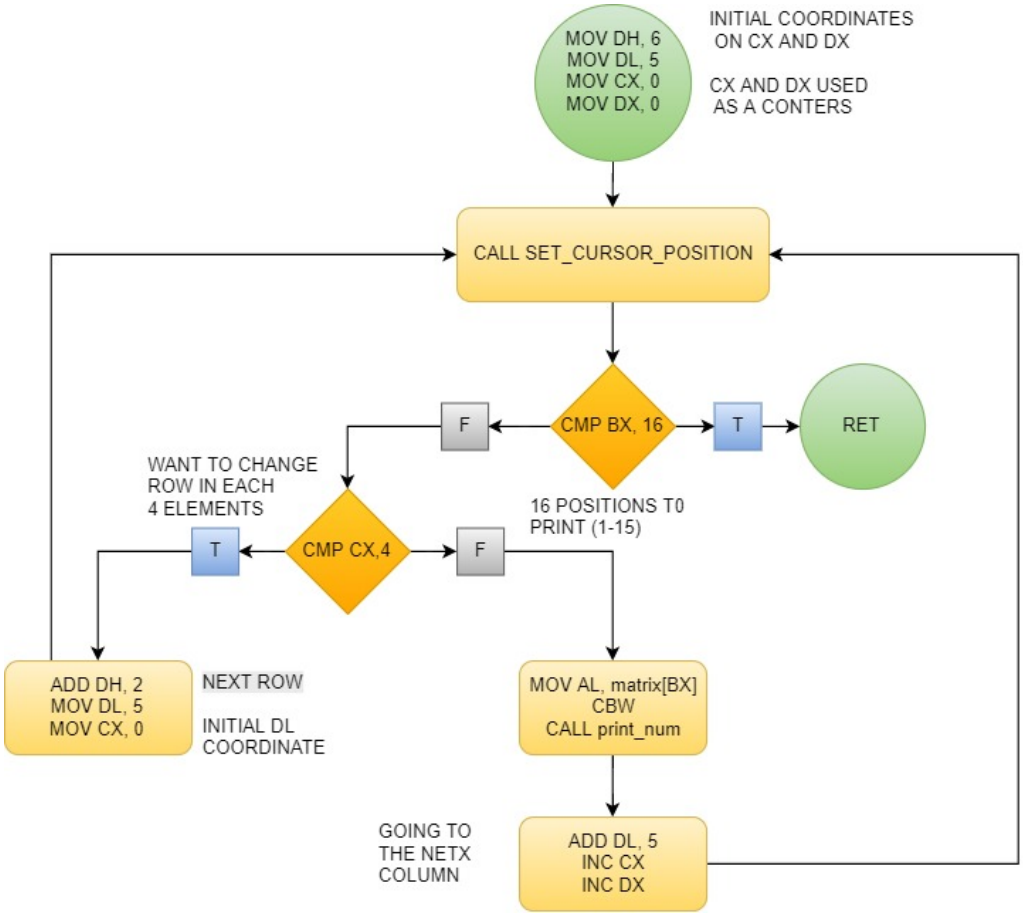


4. FUNÇÕES RELEVANTES

PRINT MATRIX

Esta função simplesmente escreve os valores da matrix nas respetivas células. Vai-se alterando a posição onde queremos escrever utilizando Set_Cursor_Position.

O funcionamente desta função é descrito em detalhe no fluxograma.



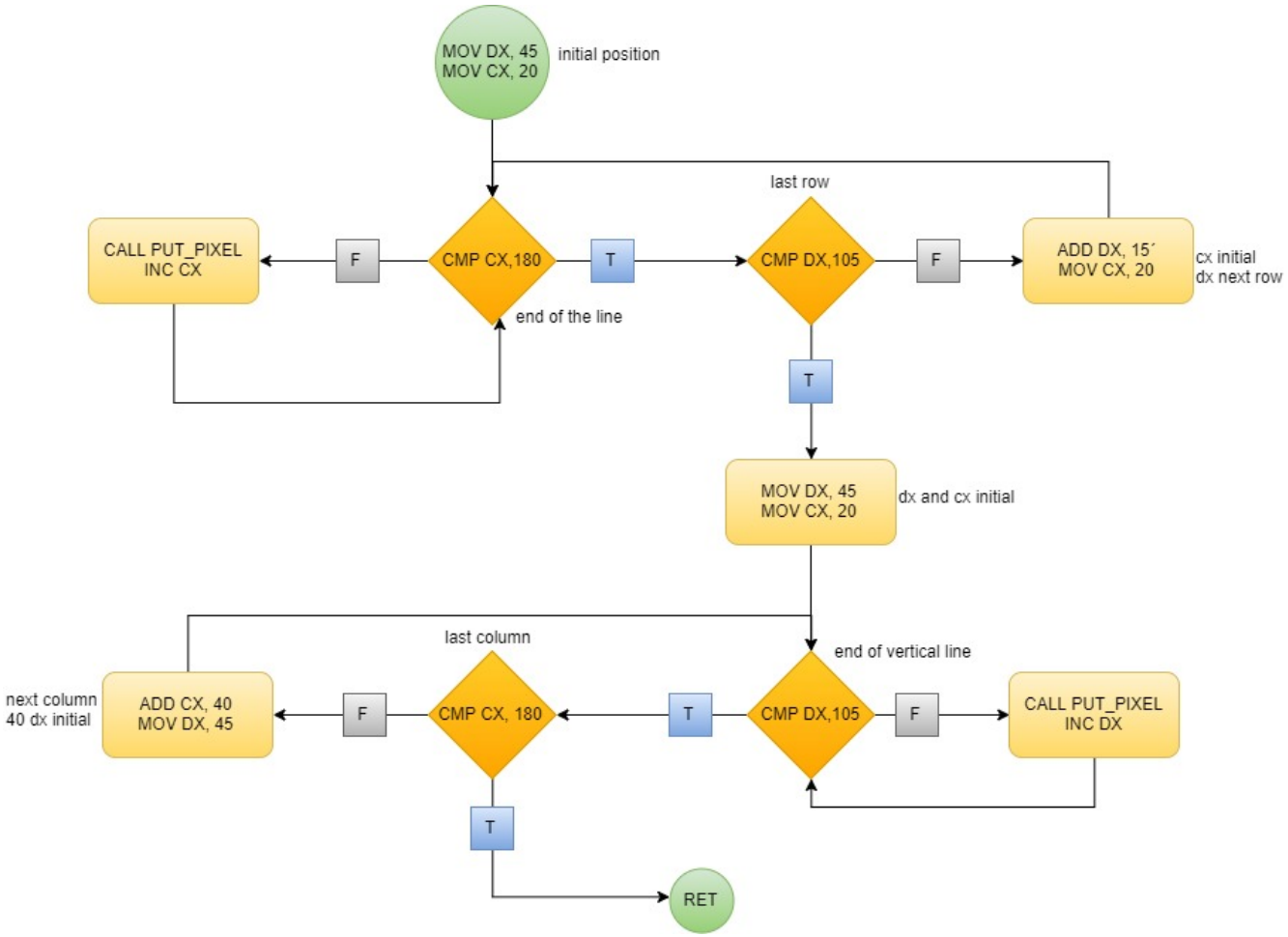
MAKE GRID

Nesta função faz-se o desenho da grid. A função recorre essencialmente à função Put_Pixel.

O funcionamento desta função está descrito detalhadamente no fluxograma.

MAKE GRID

FLUXOGRAMA



PRINT NUM UNSIGNED

Esta função imprime um número sem sinal para o ecrã.

Recebe o valor a imprimir em AX e vai divindindo por 10; a cada divisão feita é guardado o dígito respetivo ao resto no stack (PUSH) e incrementado um contador (este contador vai dar-nos o número de divisões efetuada).

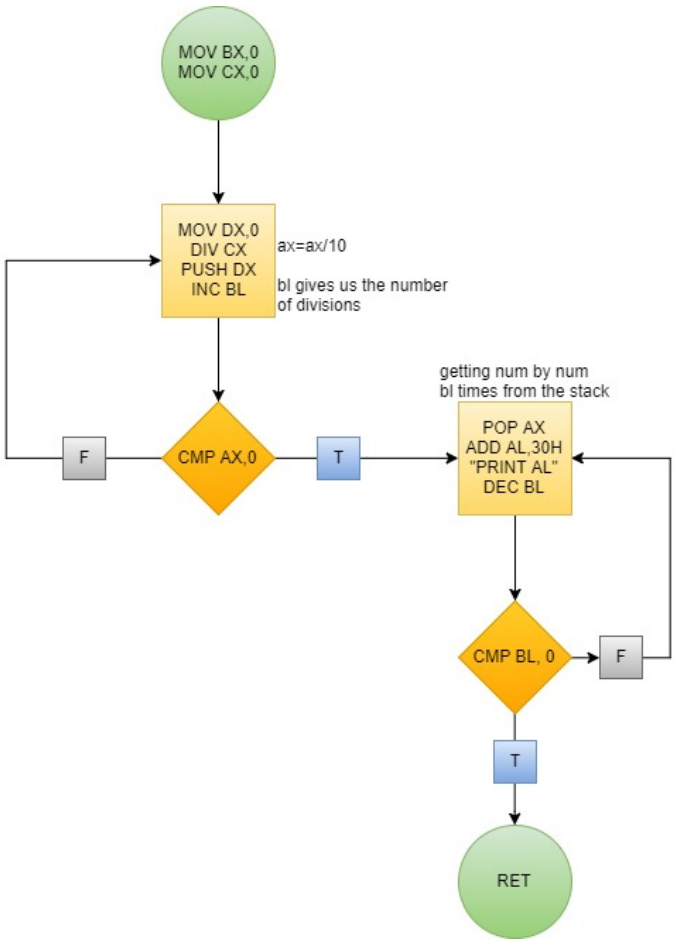
Como a sintaxe da operação que estamos a realizar é AX=AX/10 quando AX=0 paramos de dividir e fazemos a impressão do número.

Recuperamos o valor a partir do stack (POP), convertemos de ascii para inteiro e seguidamente imprimimos; realizamos este processo até o contador estar a zero.

No fluxograma fazemos uma análise mais detalhada da função.

PRINT NUM UNS

FLUXOGRAMA



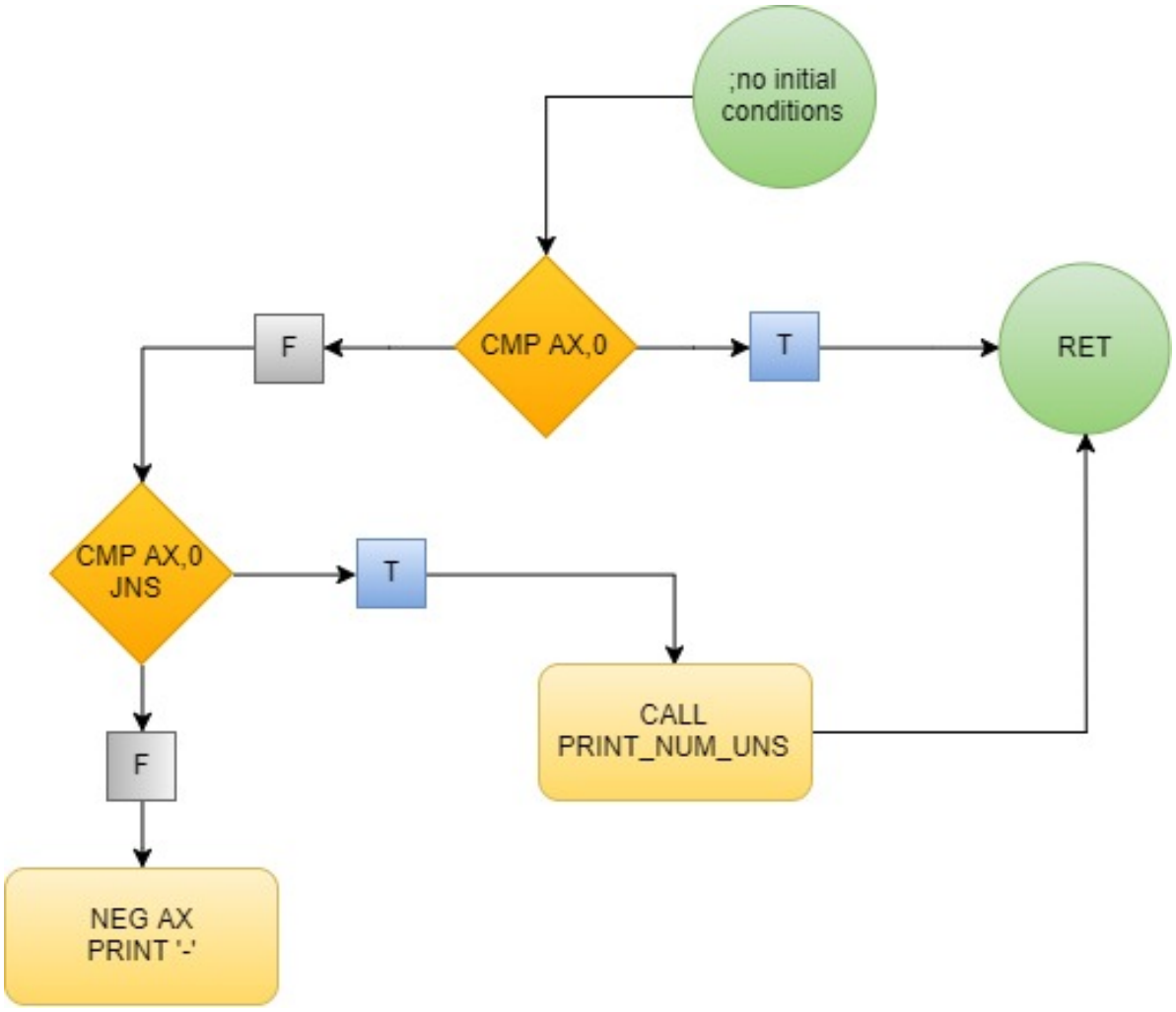
PRINT NUM (SIGNED)

Esta função tem um pequeno pormenor a mais em relação à PRINT_NUM_UNS, vê se o número a imprimir é signed, se for imprime '-' e a partir daqui tratamos o número como se fosse positivo (NEG AX); se for unsigned, chamamos logo a PRINT_NUM_UNS.

No fluxograma desta função é possível uma melhor compreensão da mesma.

PRINT NUM

FLUXOGRAMA



INT 2 ASCII UNSIGNED

Esta função faz extamente o mesmo que a função PRINT NUM UNSIGNED, contudo em vez de escrever o número no ecrã, guarda o número (carater a carater) numa string, passada para DI como parâmetro.

Esta função é utilizada no Export_Spreadsheet.

INT 2 ASCII (SIGNED)

Exatamente igual a PRINT_NUM, em vez de imprimir '-' para o ecrã, guarda na string passada a DI como parêmetro.

Esta função é utilizada no Export_Spreadsheet.

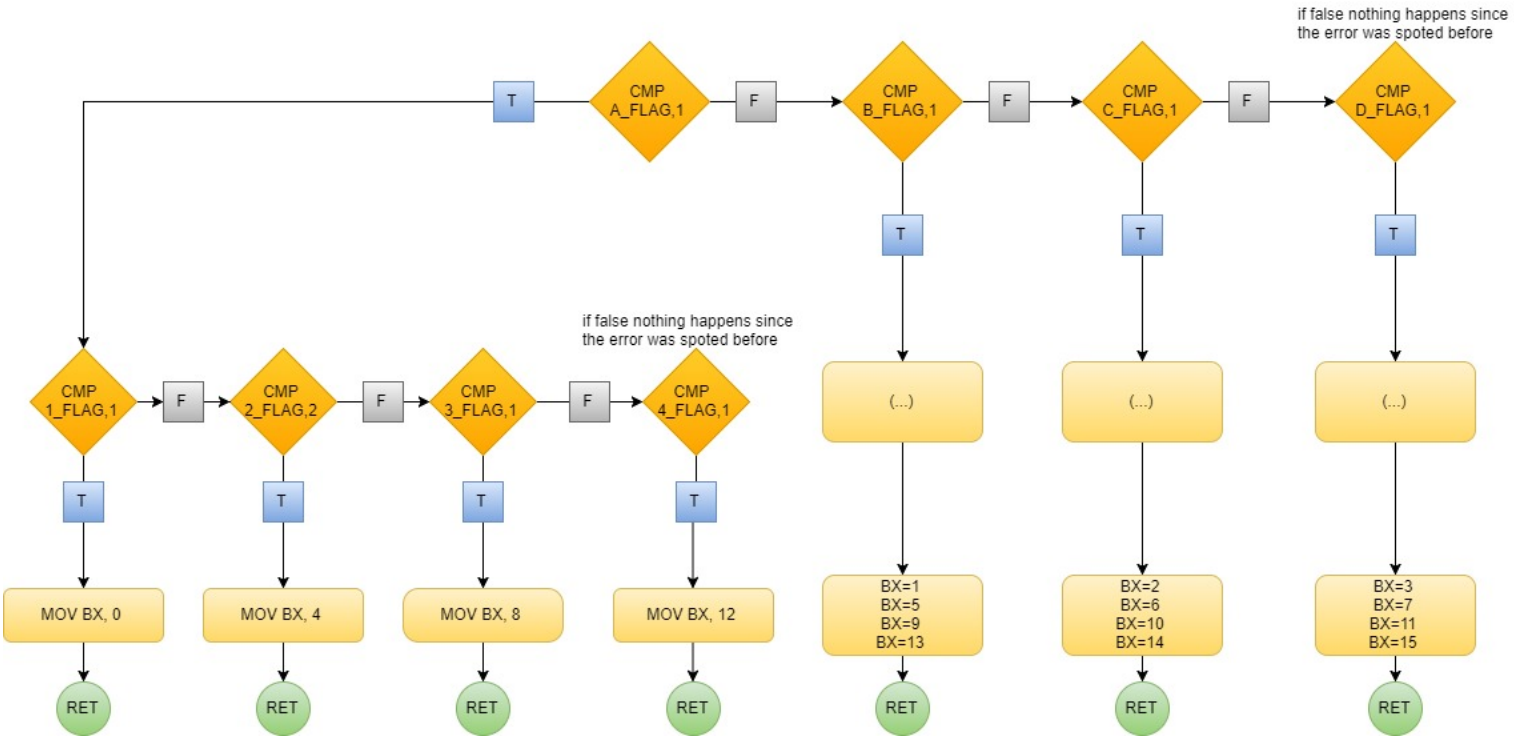
CHOOSE OPTION

Esta função analisa as flags das letras/números e consoante estas devolve a posição da matriz que deve ser alterada (quer para receber quer para transmitir a informação).

Pode percebe-se mais detalhadamente o funcionamento desta função através do seu fluxograma.

CHOOSE OPTION

FLUXOGRAMA



SCAN NUM

Lê um número inteiro (com sinal) do teclado e armazena o mesmo em CX.

Esta função tem em conta que não podem ser introduzidos valores superiores a 128*, nem valores que não sejam numéricos.

Uma vez que o valor lido do teclado vem como ascii, a primeira coisa a fazer é ver se este está nos valores compreendidos para números (entre 30h e 39h).

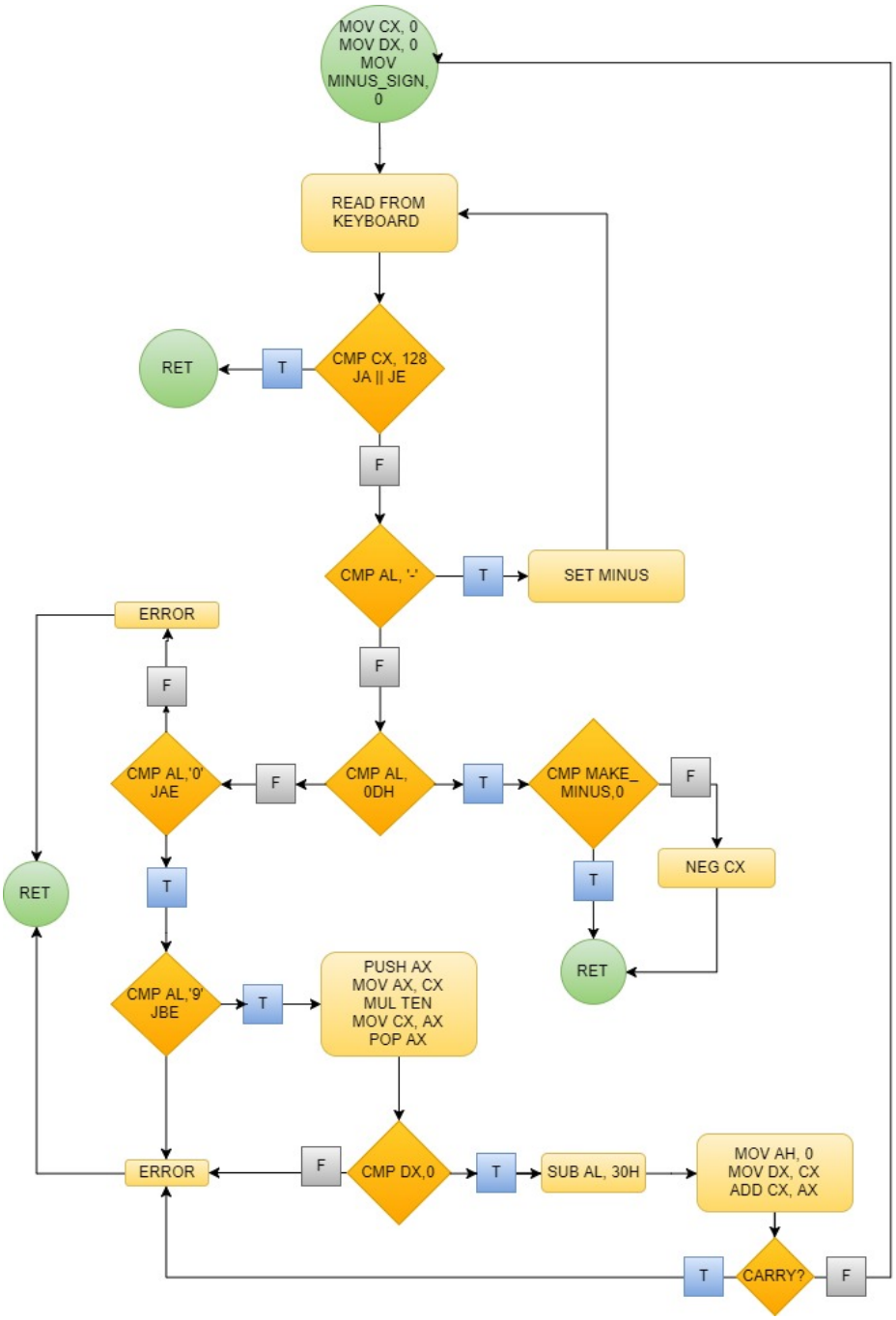
Se for introduzido o carater '-' a minus_flag é acionada e no final da execução do programa o é executada a instrução NEG.

A ideia da função é ir recebendo carater a carater e no caso de existir próximo carater multiplicamos o valor que já temos x10, convertemos de ascii para inteiroo valor introduzido (sub al,30h) e posteriormente somamos o atual com o que já tinhamos.

O funcionamento detalhado desta função é explicado no fluxograma.

SCAN NUM

FLUXOGRAMA



ASCII 2 INT

Esta função repete exatamente o mesmo processo que a SCAN_NUM, contudo, uma vez que é aplicada à leitura do número no ficheiro esta termina quando "recebe" ';' em vez de ENTER, como acontece na SCAN_NUM.

Notar também que, em vez de ler do teclado esta função espera que lhe seja passado como parâmetro para DI a string que queremos analisar.

*Nota: Porquê 128? Uma vez que estamos a trabalhar numa interface que não nos permite números acima de 16 bists para podermos imprimir números sem que ocorram erros fazemos operações utilizando somente números com 8bits (256). Uma vez que estamos a trabalhar com signed number, ainda temos que "partir" este valor ao meio.

Ou seja, chegamos ao valor 128.

CALCULATOR

Esta pequena "calculadora" simplesmente recebe dois operandos, lê a flag de operação e consequentemente decide qual a operação a realizar.

Faz a operação e devolve o resultado na variável result_num.

NOTA IMPORTANTE: infelizmente, esta calculadora é incapaz de realizar divisões com números negativos.

FORMULA READER

Esta função lê a fórmula e realiza a operação aritmética.

Caso seja introduzido pelo utilizador algo "inesperado", tudo até então será apagado e o utilizador terá que escrever a fórmula do início.

Se o utilizador desejar "sair" sem introduzir qualquer valor, deve carregar ENTER quando tudo estiver em branco, ie, se o utilizador carregar ENTER aquando introdução da fórmula, sem ser na primeira "leitura", este inserção vai ser dada como inválida.

Se tudo for introduzido corretamente vão ser acionadas as flags correspondentes aos números/letras e operação; Notar que, ao ser acionada a flag faz-se "PUSH AX" para posteriormente poder guardar-se a fórmula (FORMULA BUFFER).

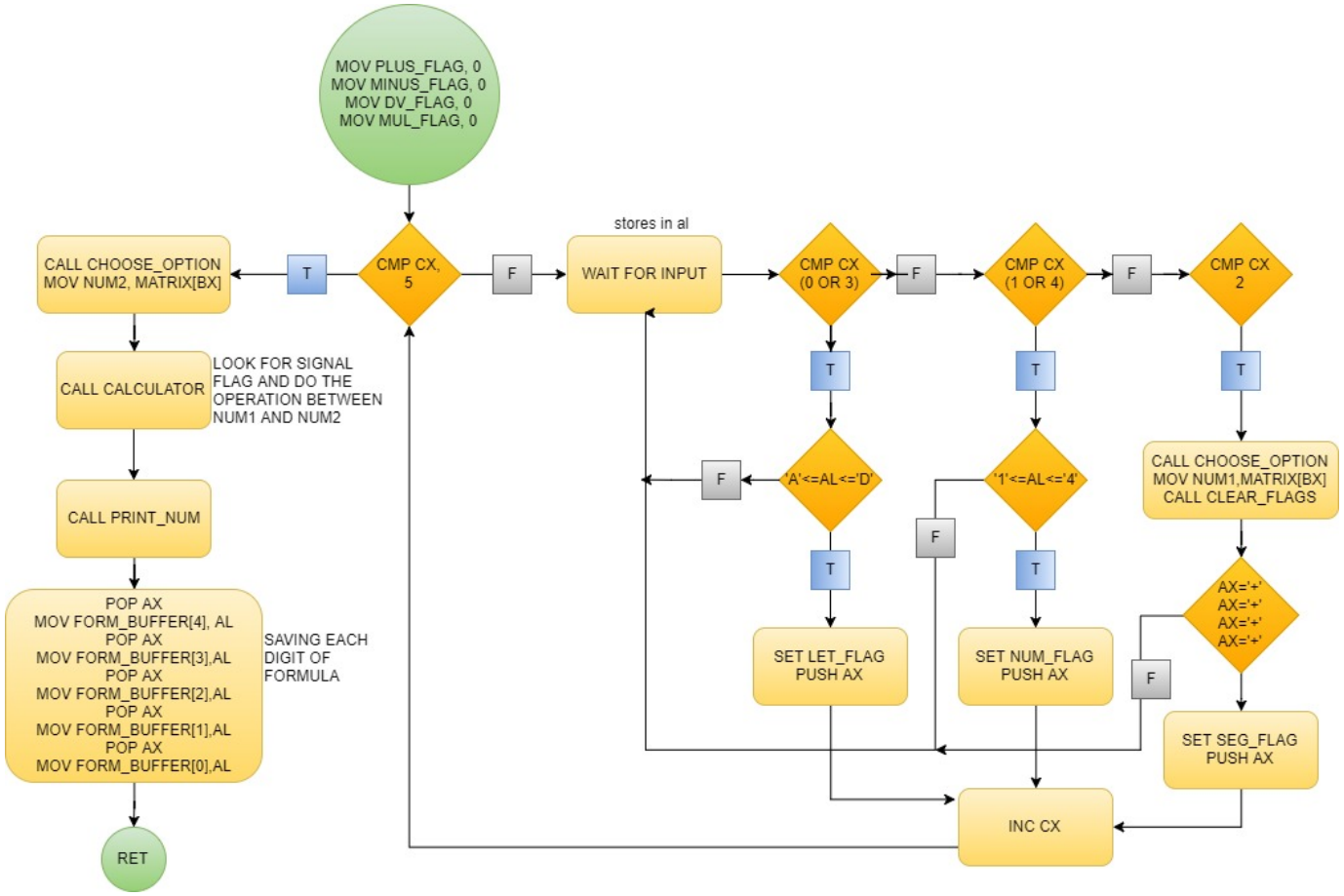
Após ter-mos os dois operandos e o "sinal" da operação a realizar chamamos a função Calculator, que nos devolve o resultado pronto a ser impresso.

Por fim guardar-se-á a fórmula no respetivo buffer (utilizando POP AX, uma vez que os valores de cada dígito foram guardados anteriormente no stack).

Podemos obter uma visão mais detalhada acerca desta função analisando o respetivo fluxograma.

FORMULA READER

FLUXOGRAMA



RESULT UPDATE

Esta função é muito idêntica à Formula_Reader; contudo não espera o input de uma fórmula.

A primeira coisa a fazer é verificar a existência de fórmula, caso não exista, a função termina.

Se existir fórmula, ativa flags e posteriormente realiza a operação.

(Informação mais detalhada na função Formula_Reader)

5. OUTRAS FUNÇÕES

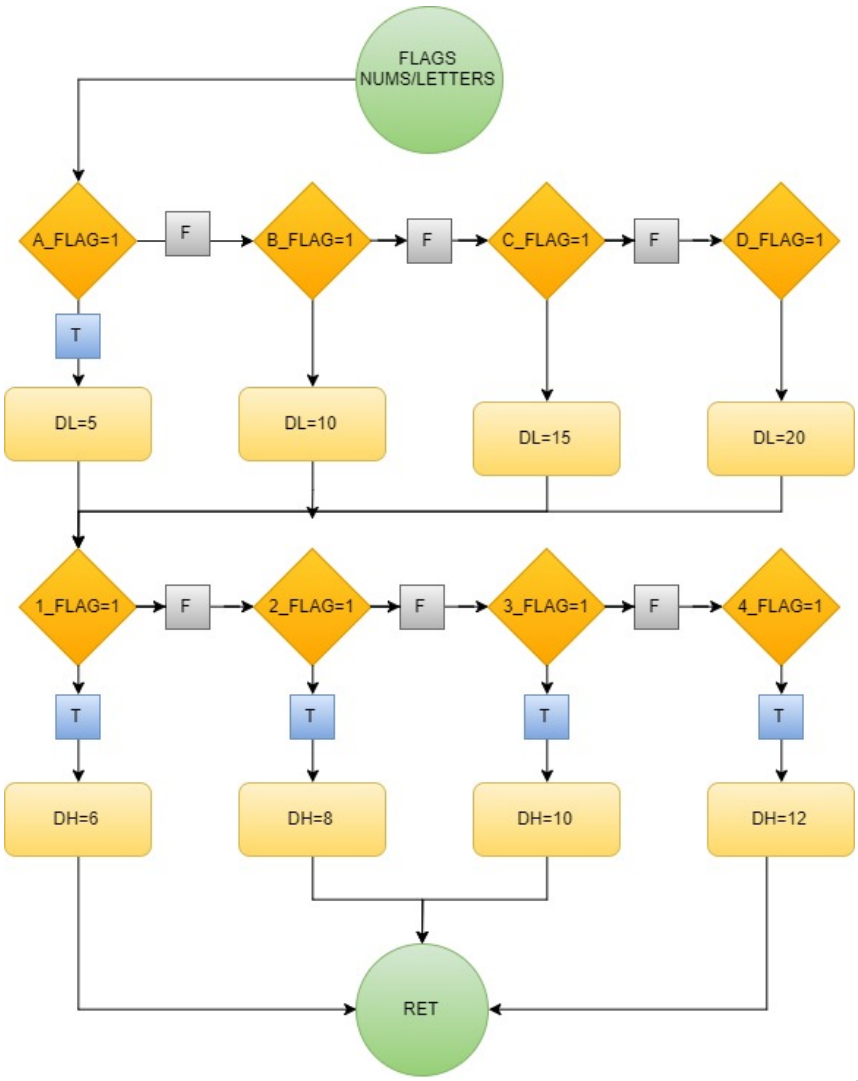
SET POSITION MATRIX

Analisa as flags de letras/números e devolve em DX as coordenadas corretas da posição na qual queremos escrever.

Set_CursorPostion lê as coordenas em DH(row),DL(column), esta funcão deixa-as neste "formato" para facilitar.

SET POSITION MATRIX

FLUXOGRAMA



READ KEYBORD

Lê uma string do teclado e guarda em DI.

Esta função é usada para receber o nome dos ficheiros.

PRINT GRID NUMS FLAGS (aka PGNL)

Esta função imprime A,B,C,D e 1,2,3,4 nas posições corretas aquando do funcionamento do Edit_Spreadsheet e Show_Spreadsheet.

PRINT RESULT AND FORMULA STRING (aka PTFS)

Faz exatamente o que diz o seu nome, imprime as strings "FORMULA", "MENU" e "RESULT" nos sítios corretos.

É utilizado aquando do funcionamento do Edit_Spreadsheet e Show_Spreadsheet.

RESET FLAGS

"Reseta" as flags de letras/números. (Não altera as flags de sinal aritmético)

CLEAR STRING

Esta função recebe como parâmetro a string que deve ser "limpa" em DI e o número de bytes em BX.

Todos os elementos da string passada ficam com o valor 0h.

MAKE BLANKS

Usamos esta função para "limpar" as células da matriz.

SET FLAGS

Recebe AL como parâmetro de entrada e analisa o seu valor; Aciona a respetiva flag.

Caso nenhuma flag seja ativada AX=-1 , se tudo correu bem e foi acionada uma flag AX=1.

6. CONCLUSÕES

Com a realização deste trabalho foi possível aprofundar alguns temas das aulas teóricas bem como vê-los “detalhadamente” em ação.

Permitiu-nos que ficássemos a conhecer o “mundo” do Assembly, que, apesar de se tratar de uma linguagem de baixo nível, se demonstrou uma poderosa e útil ferramenta.

Em suma, consideramos que o resultado final é bastante positivo e que foram atingidos os objetivos propostos.

7. REFERÊNCIAS BIBLIOGRAFICAS

Slides Microprocessadores 2017 ímpar, João Paulo Pimentão e Pedro A. C. Sousa.