

PROJECTO E IMPLEMENTAÇÃO DE UMA INTERFACE PARA TECLADO
EM VLSI

J. C. DELGADO (*) e R. P. MARTINS (**)

SUMARIO

Neste artigo descreve-se o projecto e implementação de uma interface para teclado como parte integrante do projecto de um circuito integrado destinado à marcação de um número telefónico por impulsos a partir de um teclado.

São abordados os pormenores de projecto do circuito, através da descrição dos seus blocos constituintes, descrito o processo de verificação do seu correcto funcionamento por meio da simulação lógica e, finalmente, apresentada a sua disposição no layout do circuito integrado.

A modularidade e tempo de projecto constituíram factores importantes no desenvolvimento, o que se reflecte no desenho de um número mínimo de células básicas (depois repetidas) e no recurso à lógica regular (PLAs).

(*) - Centro de Análise e Processamento de Sinais, IST.

(**) - Centro de Electrónica Aplicada, IST.

1. INTRODUÇÃO

O projecto descrito neste artigo insere-se num programa mais vasto de produção de um circuito integrado destinado a efectuar a marcação de um número telefónico por impulsos, substituindo o vulgar marcador por um teclado.

O circuito dispõe de alguns extras em relação a circuitos integrados de funções idênticas existentes no mercado, como memória RAM incorporada com capacidade para 9 números telefónicos de 26 dígitos, interface para mostrador de cristais líquidos e um contador de impulsos de contagem de períodos na rede telefónica.

Este artigo descreve a parte do circuito que realiza a interface com um teclado de matriz de contactos e reflecte o trabalho dos autores no projecto do circuito integrado.

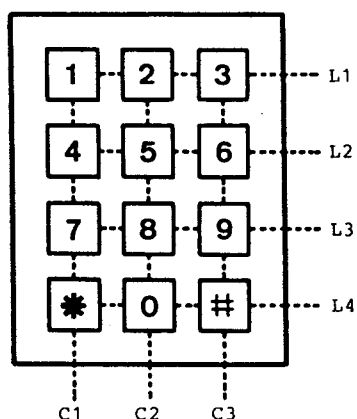


Fig. 1 - Teclado telefónico standard de 12 teclas.

A interface prevê um teclado telefónico standard de 12 teclas (figura 1), organizado em 4 linhas por 3 colunas, e as suas principais funções são as de efectuar o varrimento das teclas, para detectar se alguma foi premida, o "debouncing" dos contactos, detecção de mais de uma tecla premida e a

transcodificação entre o código da posição que cada tecla ocupa no teclado e o seu verdadeiro valor, que é depois enviado para os restantes blocos do circuito integrado.

Desta última função ressalta a tecla 0 (zero), que deve ser convertida no valor 10 para que corresponda directamente, sem mais processamento, a 10 impulsos na linha telefónica.

2. PRINCIPIOS DE FUNCIONAMENTO

O esquema de blocos do circuito está representado na figura 2.

O teclado está organizado como uma matriz de contactos, em que o premir de uma tecla estabelece um contacto entre a sua linha e a sua coluna. Para converter esta informação em 4 bits torna-se necessário efectuar um varrimento, que é conseguido colocando um "0" numa das colunas (C1 a C3 na figura 1), enquanto as outras permanecem a "1", e lendo sucessivamente o valor de cada uma das linhas (L1 a L4).

Os contadores CONT1 e CONT2, accionados pela unidade de controle (UC), formam um contador de 4 bits que irá fornecer o código da posição da tecla. Os dois bits de maior peso seleccionam, através do demultiplexer MPX2, qual das colunas terá o "0" e os dois bits de menor peso seleccionam uma das linhas, através do multiplexer MPX1, fornecendo o estado lógico dela à UC na variável KEY. O multiplexer MPX1 tem "pull-ups" internos de modo a que uma linha sem teclas carregadas seja lida como "1".

Se a tecla 1, por exemplo, for premida, só aparecerá um "0" em KEY quando os contadores CONT1 e CONT2 estiverem em

"0000", altura em que a primeira coluna e a primeira linha são seleccionadas (ver figura 1).

Para simplificar o circuito, e apesar de o teclado possuir apenas 3 colunas (12 teclas), deixa-se os contadores contar os 16 periodos, não havendo hipótese de KEY valer "0" durante os últimos 4, correspondentes à quarta coluna, inexistente.

O transcodificador é um circuito combinatório que converte o código de posição da tecla para o seu valor correcto. Por exemplo, a tecla 1 ocupa a posição "0000" no teclado e o valor a transmitir ao bus interno deve ser "0001". Este bloco obedece à seguinte tabela :

TECLA	POSIÇÃO NO TECLADO	SAÍDA DO TRANSCODIFICADOR
1	0	1
2	4	2
3	8	3
4	1	4
5	5	5
6	9	6
7	2	7
8	6	8
9	A	9
0	7	A
*	3	0
#	B	B

Note-se ainda que a tecla 0 foi convertida para 10 (A hexadecimal) para que esta tecla corresponda a 10 impulsos marcados na linha telefónica.

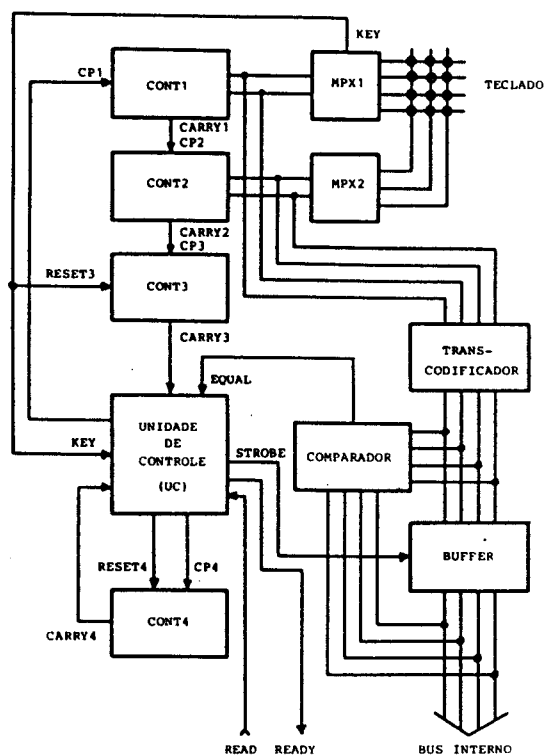


Fig. 2 - Esquema de blocos da interface de teclado.

Os contadores CONT3 e CONT4 têm como responsabilidade o "debouncing" dos contactos. Uma tecla só é reconhecida depois de :

1 - 3 varrimentos completos sem nenhuma tecla carregada, o que é indicado pelo CONT3 através de CARRY3. Note-se que o RESET3 é actuado pela variável KEY, o que quer dizer que, enquanto uma tecla estiver a ser premida, o reset do CONT3 estará a ser feito em cada varrimento. Esta condição tem como consequência o facto de, enquanto uma tecla estiver a ser carregada, o premir de outra não ter qualquer efeito.

2 - 3 varrimentos com a mesma tecla carregada, o que é indicado pelo CONT4 através de CARRY4. Isto evita que uma

tecla possa ser interpretada mais do que uma vez devido às oscilações ("bounces") no momento do contacto.

A segunda condição é garantida com a ajuda do comparador e do buffer. Sempre que descobre uma tecla premida através da variável KEY, a unidade de controle compara o seu valor (já corrigido pelo transcodificador) com o valor armazenado no buffer. Se forem iguais, incrementa o contador CONT4; se não, armazena o novo valor no buffer, activa o RESET4 e volta ao varrimento.

O funcionamento da unidade de controle torna-se mais explícito através do seu fluxograma (figura 3).

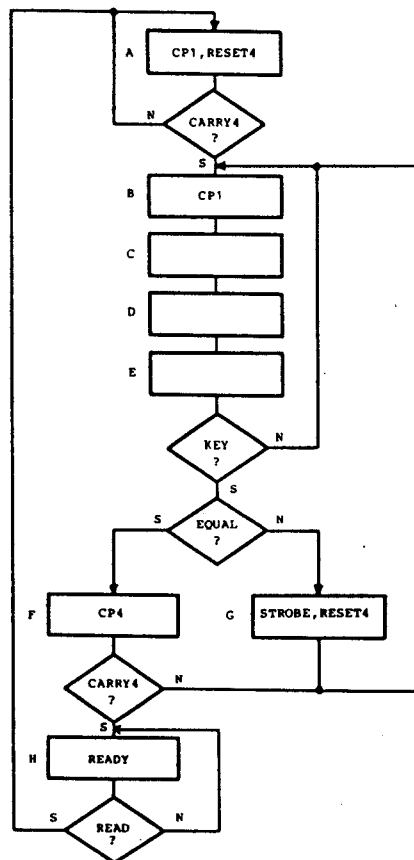


Fig. 3 - Fluxograma da unidade de controle.

Depois do reset do circuito, a unidade de controle entra no estado A, em que activa os contadores de varrimento por meio de CP1 e, acessoriamente, faz o reset do CONT4. Após 3 varrimentos sem teclas carregadas, passa ao ciclo de pesquisa de teclas (estados B a E), de que só sai quando uma tecla for premida (KEY activo, ver figura 2).

Os estados C, D e E destinam-se a reduzir a frequência de varrimento, pelo que só no estado B os contadores de varrimento são incrementados. O circuito funciona com um relógio de duas fases $\phi 1$ e $\phi 2$ com uma frequência de cerca de 10Khz (período 100ns). Os 4 estados B a E demoram assim cerca de 400ns e o varrimento completo 16 vezes mais (ciclo de CONT1 e CONT2), ou seja, cerca de 6.4ms.

Os 3 varrimentos completos exigidos quer para o intervalo mínimo entre teclas quer para a duração mínima do premir de uma tecla correspondem a 20ms, valor perfeitamente aceitável para os teclados usuais.

Quando for detectada uma tecla premida, o valor presente na saída do transcodificador é comparado com o valor já armazenado no buffer, sendo o resultado da comparação comunicado à unidade de controle através da variável EQUAL (figura 2).

Se forem diferentes, o novo valor é colocado no buffer (activação do STROBE) e é feito o reset do contador CONT4 (estado G); se forem iguais, então o CONT4 é incrementado activando CP4 (estado F). Se este chegar ao fim da sua contagem (o CARRY4 aparece ao fim de 3 impulsos em CP4 depois do último reset), a tecla é reconhecida por ter estado premida 3 varrimentos e o sinal de READY (destinado à unidade de controle de todo o circuito integrado) é activado.

A unidade de controle desta interface mantém-se no estado H até que seja feita uma leitura do valor da tecla através do sinal de READ, altura em que retoma o ciclo de varrimento.

3. PROJECTO

Dada a baixa frequência do relógio (cerca de 10Khz), decidiu-se optar por registos estáticos. Como consequência, o espaço ocupado é maior do que o correspondente à implementação dinâmica, mas o funcionamento é mais seguro.

A célula básica de um registo de deslocamento dinâmico está representada na figura 4 e a correspondente estática na figura 5. Como se pode ver, esta última é mais complexa mas funciona em frequências tão baixas quanto se quiser.

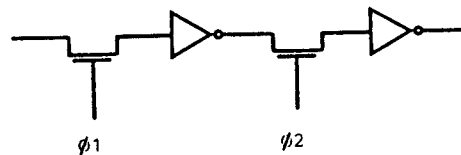


Fig. 4 - Célula básica de um registo dinâmico.

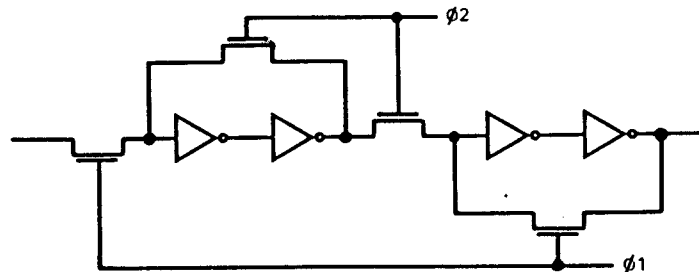


Fig. 5 - Célula básica de um registo estático.

O circuito dos contadores é praticamente o mesmo para todos eles, com excepção de alguns pormenores de implementação. Para obter um circuito mais simples, os contadores contam em código reflectido, isto é, a sua sequência de contagem é 00,01,11,10.

Nos contadores CONT3 e CONT4 isto não tem qualquer importância, pois só interessam as saídas de carry. Nos contadores CONT1 e CONT2, a consequência é o varrimento ser feito também em código reflectido, mas o importante é que todas as teclas sejam pesquisadas, e não a ordem pela qual o varrimento é feito.

Assim, o esquema (algo simplificado) de um contador de dois bits é o da figura 6, em que cada registo é o circuito da figura 5.

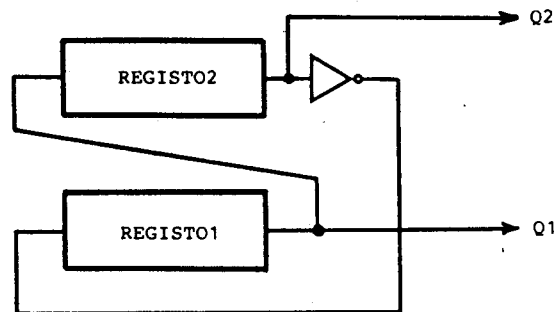


Fig. 6 - Esquema básico de um contador de 2 bits em código reflectido.

Na realidade, a negação nem é necessária, desde que a realimentação para o registo 1 seja feita a partir do ponto médio do último par de negações da figura 5.

O circuito básico dos multiplexers está representado na figura 7. O MPX2 serve na realidade de demultiplexer, tendo o ponto comum ligado à massa de modo que possa injectar um "0"

numa das colunas. O valor lógico "1" nas colunas não seleccionadas é garantido por "pull-ups".

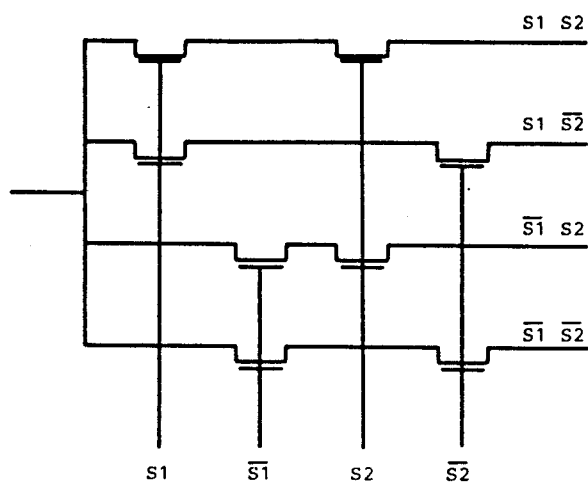


Fig. 7 - Circuito básico do multiplexer.

O transcodificador, circuito puramente combinatório, obedece à tabela já apresentada na secção 2 e consiste apenas numa PLA de 4 entradas por 4 saídas.

O comparador é feito à custa da repetição de um célula básica de comparação de um bit, que consiste essencialmente num ou exclusivo, de acordo com a figura 8.

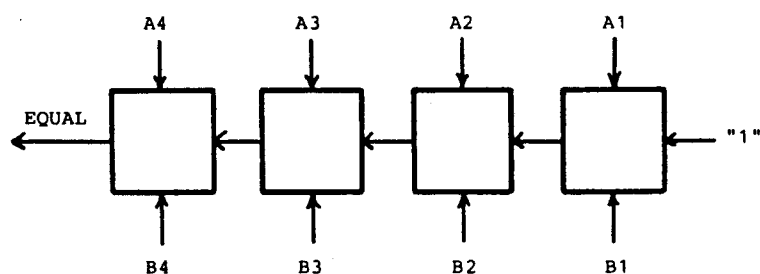


Fig. 8 - Comparador.

O buffer é a justaposição de 4 registos estáticos do tipo do da figura 5, em que as fases $\phi 1$ e $\phi 2$ se fazem depender da

variável STROBE para que o registo só desloque no ciclo do relógio em que o STROBE estiver activo.

Finalmente, a unidade de controle é constituída por uma PLA de 8 entradas (5 entradas de variáveis de controle + 3 bits para a máquina de estados - ver figura 3) por 8 saídas (5 sinais de controle + 3 bits para a máquina de estados).

No que respeita à máquina de estados, incluíram-se três células do tipo do da figura 5 entre os três bits de saída da PLA para a máquina de estados e os correspondentes bits de entrada. O registo da figura 5 foi aqui ligeiramente modificado com a substituição da segunda negação em cada par por uma gate NOR, afim de acomodar o sinal de reset do circuito, necessário à inicialização da máquina de estados da unidade de controle.

4. SIMULAÇÃO LÓGICA

Para verificar o correcto funcionamento do circuito descrito, este foi completamente simulado em computador utilizando o simulador lógico do INESC. Este simulador funciona ao nível das funções lógicas básicas (nand, nor, etc.) e permitiu descobrir alguns erros que teriam passados despercebidos sem esta ferramenta indispensável.

As entradas e saídas do circuito são os sinais de protocolo entre a interface do teclado e a unidade de controlo de todo o circuito integrado (READ e READY), o bus interno (de 4 bits) e as 12 teclas. Estas foram introduzidas no simulador substituindo o contacto de cada tecla por um transistor e considerando a gate do transistor como uma entrada lógica,

equivalendo o premir de uma tecla à activação desta entrada (figura 9).

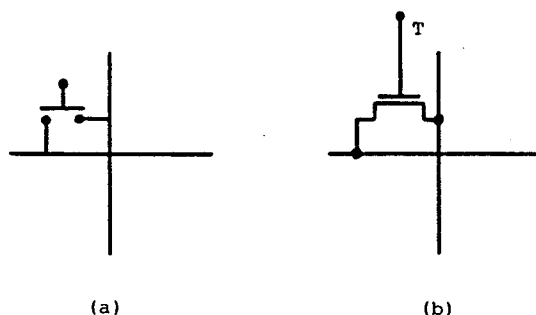


Fig. 9 - Introdução das teclas no simulador.

(a) - Contacto de um teclado.

(b) - Transistor que simula o premir da tecla quando $T = "1"$.

A simulação consta de dois ficheiros, um em que é introduzida a descrição do circuito e outro em que é especificada a sequência das entradas do circuito ao longo do tempo.

Neste último houve o cuidado de activar cada variável de entrada correspondente a cada tecla o número de períodos de relógio suficiente para a tecla ser reconhecida e deixar entre teclas o intervalo necessário.

A simulação consistiu em carregar nas teclas (activar as entradas correspondentes) uma por uma, verificar se a saída de READY era actuada, activar o sinal de READ e verificar se o valor presente à saída para o bus interno era o correcto para cada tecla.

Situações anómalas, como várias teclas premidas simultaneamente e premidas sem o intervalo necessário ou durante um tempo insuficiente foram também investigadas, verificando-se assim o funcionamento correcto do circuito.

5. IMPLEMENTAÇÃO

Depois da simulação lógica passou-se à fase de desenho do layout do circuito, representado esquematicamente na figura 10. No topo aparecem 4 pads de entrada e do lado esquerdo 3 pads de saída, destinados à ligação às linhas e colunas do teclado, respectivamente.

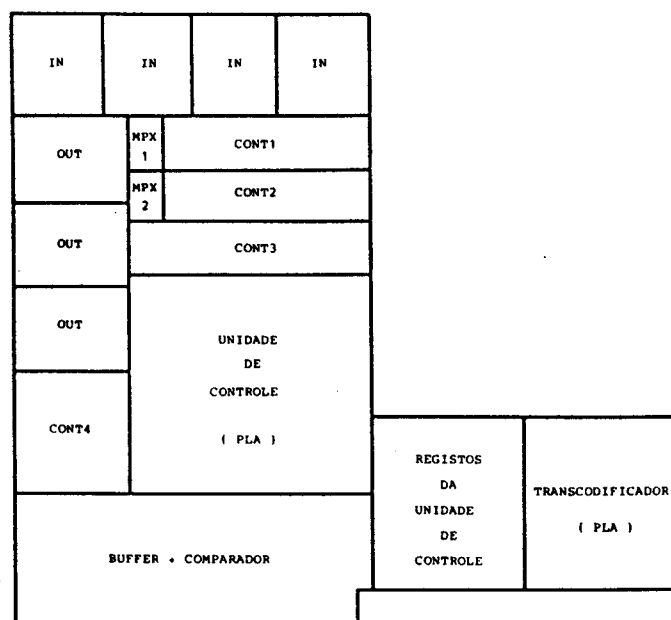


Fig. 10 - Layout esquemático da interface do teclado.

Pretendeu-se reduzir o tempo de projecto o mais possível, através do desenho de um número mínimo de células básicas e recurso à lógica regular (PLAs).

Verificou-se, no entanto, e apesar da disposição criteriosa dos blocos básicos, que o trabalho de routing das ligações absorveu a maior parte do tempo, o que reflecte a utilidade de um editor gráfico minimamente poderoso e que o INESC até à data não possui.

Na figura 11 representa-se o aspecto do layout, tal como aparecerá na chip, e na figura 12 mostra-se a localização

desta interface do teclado na área total da chip, com a identificação dos blocos mais importantes.

A chip tem cerca de 5.08x5.08 mm e será fabricada em tecnologia NMOS de 2.5 μm .

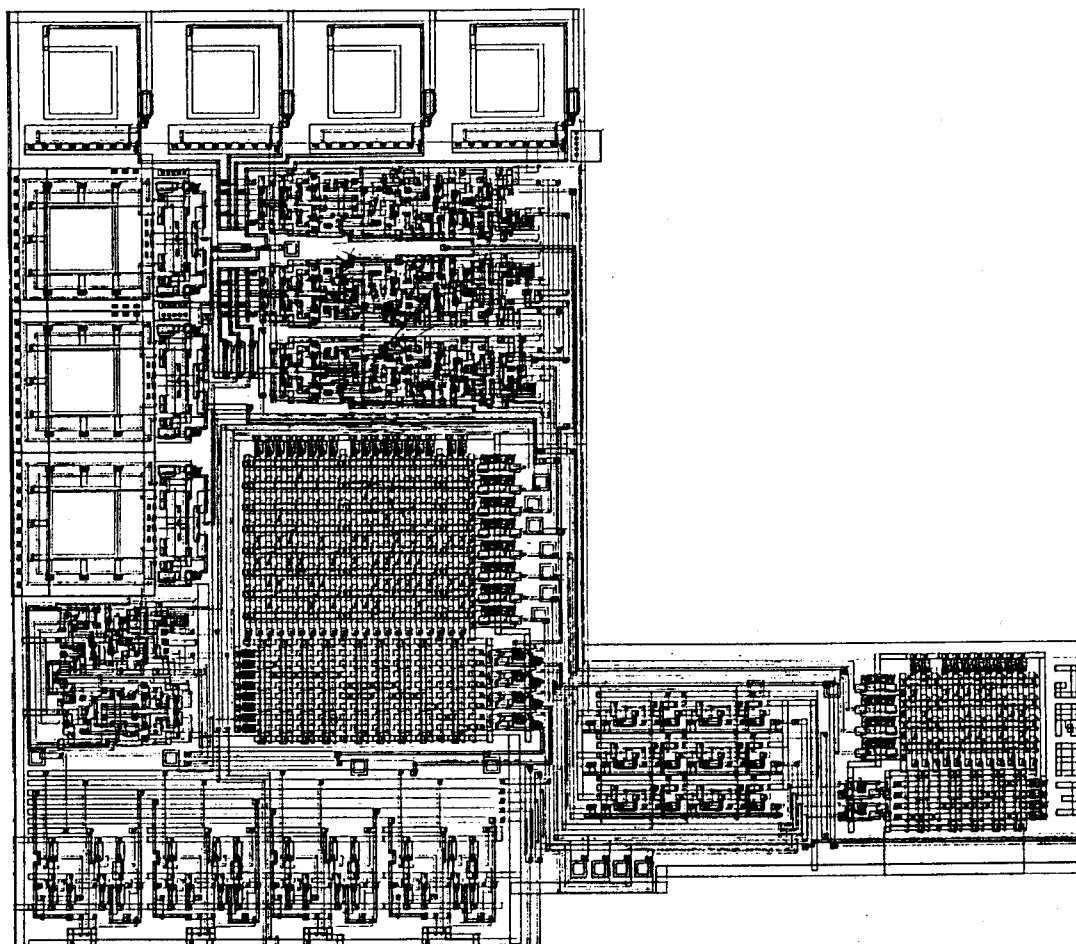


Fig. 11 - Layout da interface do teclado.

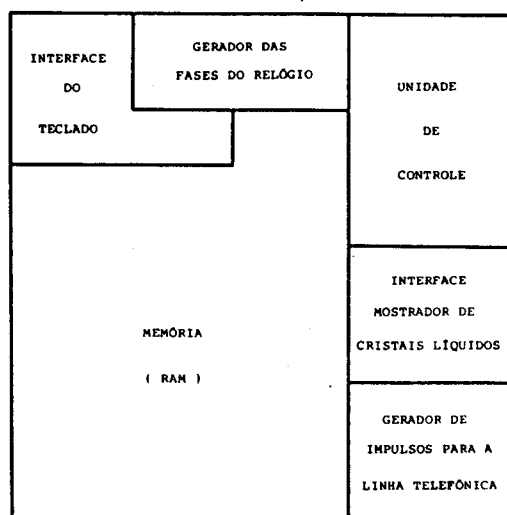


Fig. 12 - Layout esquemático da chip total.

6. CONCLUSÕES

Descreveu-se neste artigo a experiência vivida pelos autores no projecto e implementação de um circuito de VLSI de marcação de números telefónicos através de um teclado.

Este projecto tem essencialmente um carácter experimental, sendo de longe a chip mais complexa já projectada em Portugal, e reflecte o trabalho de coordenação de 17 pessoas, número bastante elevado para um projecto deste tipo. Nomeadamente, a definição dos protocolos de comunicação entre blocos e atribuição de área aos diversos circuitos com vista à maior compactação possível constituíram os maiores desafios do projecto.

É convicção dos autores que o trabalho dispendido foi extremamente positivo, não só pelos resultados alcançados e experiência adquirida mas também pela definição, ao longo do projecto, dos métodos de trabalho adequados e das ferramentas necessárias a um projecto desta natureza.

AGRADECIMENTOS

O trabalho descrito neste artigo foi suportado pelo INESC. Os autores agradecem a orientação prestada pelos Profs. Luis Vidigal e Alves Marques e pelo Eng. António Leal e as discussões mantidas com todos os colegas de projecto, que no seu conjunto tiveram uma influência decisiva no resultado final.

BIBLIOGRAFIA

C. A. Mead e L. Conway, "Introduction to VLSI Systems", Addison Wesley, Reading, Mass. 1980.