

Formas como PLDs são Programados

Guilherme Evangelista de Albuquerque
Instituto Federal de Santa Catarina
Campus São José
guilherme.ea@sj.ifsc.edu.br

Tiago Teixeira
Instituto Federal de Santa Catarina
Campus São José
TiagoTeixeira22@yahoo.com.br

19 de Fevereiro de 2015

Resumo

Neste artigo é apresentado, com intuito didático, a pesquisa de como as memórias SRAM e Flash, comumente denominada de E²PROM flash, são programadas e aplicadas aos PLDs (*programmable logic devices*), mais especificamente CPLD e FPGA. Será explicado e demonstrado como é a estrutura interna de cada uma das duas memórias e como cada uma de suas células são programadas/apagadas, e também será explicado como é a parte lógica dos dois PLDs e como essas duas memórias são aplicadas aos PLDs.

Palavras-chavess: SRA. Flash. PLD. CPLDs. FPGA. HCPLDs.

Introdução

Dispositivos lógicos programáveis ou PLDs (*programmable logic devices*) são dispositivos utilizados para implementar funções lógicas e possibilitam. Os dispositivos PLDs são programados via software fornecidos pelo próprios fabricantes. Por PLD se entende que são circuitos integrados que não possuem função específica e que são configurados pelo usuário.

Os PLDs simples usam tecnologia CMOS e oferecem elementos de memória do tipo EPROM, E²PROM, e memória flash, e englobam os PALs, GALs, e outros componentes. Podem ser classificados em função de portas lógicas que comportam, tendo os SPLDs e os HCPLDs. Os SPLDs são dispositivos simples e de baixa capacidade, que tipicamente contêm menos de 600 portas lógicas, fabricados com tecnologia CMOS. Já os HCPLDs (dispositivos lógicos programáveis de alta capacidade), oferecem mais de 600 portas disponíveis; os mais modernos podem atingir milhões de portas disponíveis e também utilizam tecnologia CMOS com memória EPROM, E²PROM, FLASH, SRAM, e opções anti-fusível. Por HCPLDs entendem-se os dispositivos FPGA e CPLDs, que serão os dispositivos explorados nesse trabalho. Esses dispositivos utilizam as memórias FLASH e SRAM para armazenarem suas programações.

1 SRAM (Static Random Access Memory)

A SRAM (memória estática de acesso randômico) é um tipo de memória de acesso aleatório que mantém os dados armazenados desde que seja mantida sua alimentação. As memórias SRAM não precisam que as células que armazenam os bits sejam constantemente atualizadas (refresh), como acontece nas memórias DRAM. O maior problema dessa tecnologia é o preço, porque a SRAM precisa de mais transistores para operar, resultando em pentes com menos capacidade de armazenamento do que uma DRAM do mesmo preço. Uma aplicação popular da SRAM é na construção de memórias cache para computadores.

Uma implementação comum de célula SRAM é a denominada célula 6T (seis transistores). Consiste em dois inversores CMOS ligados em anel, mais dois transistores nMOS para controle de acesso, responsáveis por ligar os inversores às linhas de bits (bit lines, BLs) de entrada e saída quando a linha de palavra (word line, WL) correspondente é ativada. A programação de uma célula SRAM pode ser realizada através do sistema ISP (*in system programming*).

1.1 Procedimento de Leitura

Um dos métodos para ler a célula SRAM consiste em primeiro pré-carregar ambas as BLs até Vdd e então deixá-las flutuando. Em seguida, WL é ativada, fazendo com que a tensão de uma das BLs seja diminuída. Outra alternativa de leitura é pré-carregar as BLs até Vdd/2 em vez de até Vdd, o que impede a corrupção de dados mesmo que os transistores não estejam dimensionados adequadamente, e tem a vantagem adicional de provocar uma variação menor nas tensões das BLs, reduzindo assim o consumo de energia e melhorando a velocidade (à custa da margem de ruído).

1.2 Procedimento de Escrita

A operação de escrita em uma célula SRAM é realizada ajustando os bitlines ao valor desejado e capacitando os transistores a controlar os nós internos da célula. Para que seja possível escrever valores em uma SRAM, é preciso que os transistores estejam configurados corretamente durante o tempo de escrita (T_{wl}) para que se tenha o sinal gravado na célula. Considerando um tempo de escrita de um valor qualquer (T_w), caso $T_w \neq T_{wl}$ não seja obedecido, resultará em uma falha de escrita e o valor não será gravado com sucesso.

1.3 Técnica de Configuração do FPGA por SRAM

Nessa tecnologia, a chave de roteamento ou comutador é um transistor de passagem ou um multiplexador controlado por uma memória estática de acesso randômico SRAM. Basicamente, existem duas aplicações para as células de SRAM: controlar os nós das portas dos transistores de passagem e controlar as linhas de seleção do multiplexador que alimenta as entradas dos blocos lógicos

Devido a volatilidade dessas memórias, os FPGAs que se utilizam dessa tecnologia precisam de uma memória externa tipo FLASH ou EEPROM. Essa tecnologia ocupa muito espaço no circuito integrado, porém é rapidamente reprogramável.

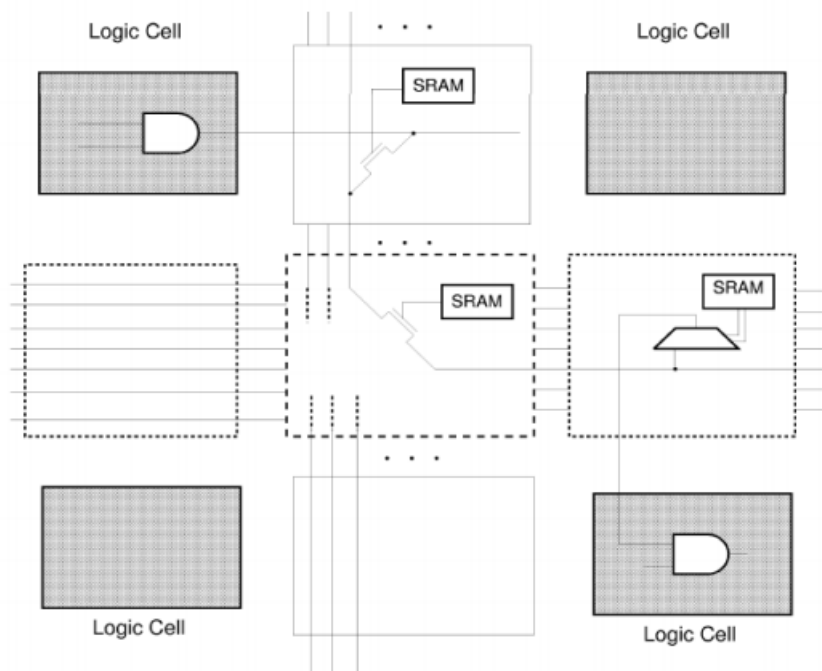


Figura 1: Chaves programáveis controladas por SRAM

2 Flash

A memória flash, também conhecida como memória E²PROM flash, é a combinação entre as memória EPROM e E²PROM. A célula flash requer somente um transistor (como a EPROM) e é eletricamente apagável/programável (como a E²PROM). Da mesma maneira que a célula da memória SRAM, a célula flash pode ser programada através do sistema ISP (*in system programmih*). Devido a sua alta densidade e baixo custo de produção é amplamente utilizada para armazenamento não volátil reprogramável.

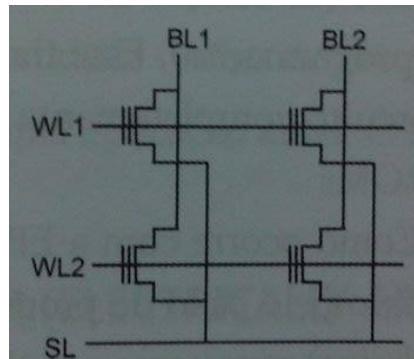


Figura 2: Célula flash

2.1 Célula ETOX

A transistor original da memória flash é o transistor ETOX (EPROM *tunnel oxide*). Duas versões do ETOX (para tecnologias de 180 nm e 130 nm) funcionam com injeção por avalanche para programação (como EPROM) e tunelamento de *Fowler-Nordheim* para apagamento (como na E²PROM).

Para programar um arranjo da memória flash, primeiro é necessário apagá-lo. O apagamento pode ser executado com pulsos negativos de tensão (-12 V^1) aplicados a todas as portas (WLs) com as fontes (SL) aterradas, ou pode ser realizado da maneira contrária, ou seja, com pulsos positivos aplicados às fontes (SL) com as portas (WLs) aterradas, sempre com os drenos (BLs) abertos, o que força os elétrons a deixarem a porta flutuante em direção à fonte (indicado como *tunnel-out* na figura abaixo). Após o processo de apagamento, todas as células flash que foram apagadas contêm o bit '1'.

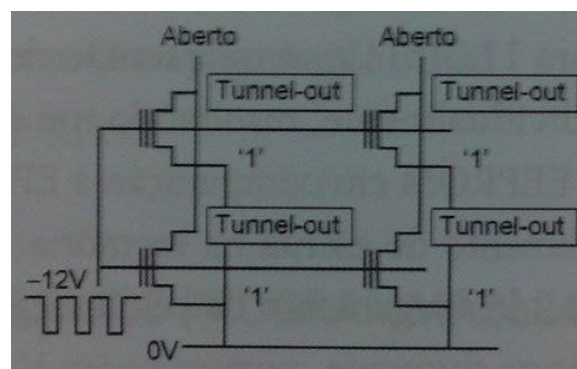


Figura 3: Célula flash - apagamento

Na programação os bits '0' e '1' são escritos na primeira linha do arranjo da memória flash. Uma tensão positiva (6 V) é aplicada na à BL1 para escrever o '0', enquanto isso a BL2 recebe uma tensão de 0 V para manter a célula apagada. A porta WL selecionada é então pulsada alta (12 V), enquanto as outras WLs em 0 V. Esse impulso provoca a injeção por avalanche dos elétrons do canal ao gatilho flutuante do primeiro transistor

¹Nota: Para o apagamento e programação da célula ETOX são necessários várias seqüências complexas de pulsos, com tensões diferentes. Essas seqüências dependem de valores particulares de tais dispositivos, variando de uma geração a outra ou mesmo de um fabricante a outro. Logo, os valores citados acima são meramente ilustrativos.

da primeira linha (indicado pelo). A segunda célula da primeira linha, que deve se manter apagada, está sujeita a um tunelamento fraco para dentro, que provoca apenas uma pequena variação da tensão daquele transistor.

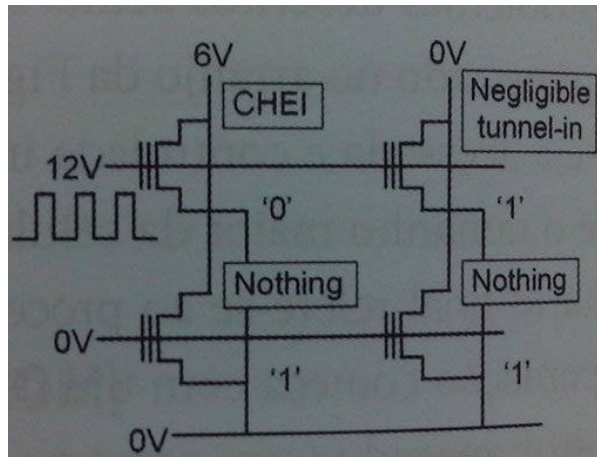


Figura 4: Célula flash - programação

2.2 Célula com Gatilho Partido

Chips independentes de memória flash geralmente possuem um microcontrolador associado que controla os procedimentos de apagamento e programação. Mesmo não causando problemas em tais chips (devido que a área de silício requerido pelo microcontrolador seja pequena), ainda sim não é adequado para sistemas que utilizam memória flash embutida.

Uma solução simples para memória flash embutida é a utilização da célula de gatilho partido (split-gate cell), como pode ser visto na Figura 5. Nesse caso, a porta flutuante cobre apenas parte do canal, enquanto porta de controle abrange todo o canal. Isso é equivalente a ter dois transistores em série para controlar a porta, sendo uma porta flutuante e outro apenas um transistor MOS comum. A desvantagem dessa é seu tamanho, não sendo adequado para células grandes. Porém, ela impede o problema do **apagamento excessivo**², elimina também a necessidade de um microcontrolador embutido com a célula, sendo ideal para células flash compactas.

2.3 Célula SONOS

A célula SONOS não possui porta flutuante, em vez disso, ela possui uma camada de nitreto de silício, que tem por função agir como armadilha de carga, portanto, quando elétrons alcançam a camada da célula, ficam presos indefinidamente (como na porta flutuante). Uma das vantagens é que ela pode ser operada com tensões menores, logo reque menos energia para ser programada. Essa tecnologia também foi aplicada à arquitetura de gatilho partido.

2.4 Flash NOR e Flash NAND

Ao contrário das arquiteturas de memórias não voláteis, que geralmente são do tipo NOR, as memórias flash são em ambas configurações, NOR e NAND. Na Flash NAND é possível observar duas linhas de seleção diferentes, denominadas seID (selecionar dreno) e seIS (selecionar fonte), elas são necessárias porque as BLs são compartilhadas por muitos módulos NAND. Tais módulos são construídos com 16 transistores cada, portanto as relações transistor por bit são 1T/1bit para NOR e 18T/16 bit para NAND.

A flash NOR normalmente é utilizada uma combinação de *CHEI* e tunelamento. Por outro lado, para flash NAND, em geral só o tunelamento é emprego, o que reduz o consumo de potência e blocos maiores sejam

²apagamento excessivo: um número excessivo de elétrons seja removido das portas flutuantes que, então, tornam-se positivamente carregadas.

processados. Nessa células, o procedimento de apagamento continua até que as células tornem-se excessivamente apagadas.

Uma das vantagens da flash NAND para flash NOR é seu menor custo devido ao fato de que os transistores na arquitetura NAND estão conectados uns aos outros, pois reduz substancialmente a região de silício. Outra vantagem é o seu menor tempo de escrita, devido ao menor tempo ao apagamento ser mais rápido. Porém, a leitura é mais lenta por causa da elevada constante de tempo associada à longa pilha de transistores.

As características de ambas fazem das memórias NAND e NOR algo mais parecido com tecnologias complementares do que concorrentes. Por exemplo, o código de programa em aplicações para computador pode não alterar muito, mas a leitura precisa ser rápida, fazendo com que a flash NOR seja adequada. Em outras aplicações como de vídeo e áudio, que exigem grandes blocos de dados sejam armazenados e renovados frequentemente, a flash NAND podem ser mais apropriadas. Um último comentário sobre as células flash, também válido para E²PROM. A durabilidade é da ordem de 10⁵ a 10⁶ ciclos de apagamento e programação.

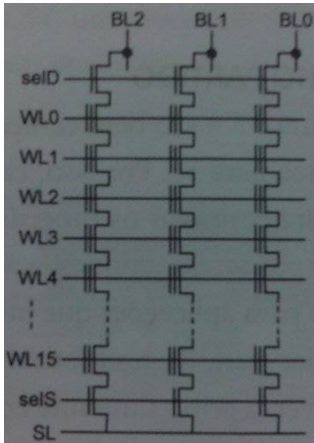


Figura 5: Flash tipo NAND

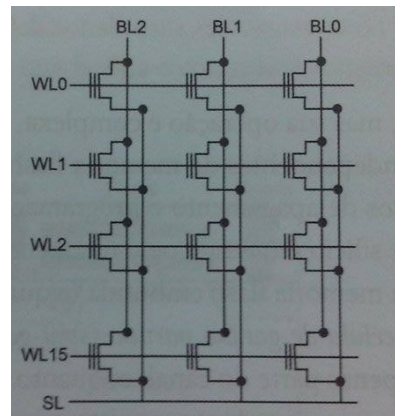


Figura 6: Flash tipo NOR

2.5 Flash Multibit

Para aumentar a densidade de dados, existem células flash de dois bits. Uma das abordagens que são utilizadas é denominada de célula *multinível* (MLC).

A célula MCL é basicamente igual a célula ETOX, porém com a capacidade de fornecer dois níveis de tensão. Neste caso, há três níveis de programação (“00”, “01”, “10”) e um nível de apagamento (“11”). A célula funciona com tunelamento e injeção por avalanche (*CHEI*) para apagamento e tunelamento para programação.

A segunda célula, denominada MBC, é semelhante à célula SONOS. Assim como na célula SONOS, os elétrons permanecem na região que foram capturados. Portanto, o bit armazenado no lado esquerdo dessa célula é independente do bit armazenado no lado direito da célula.

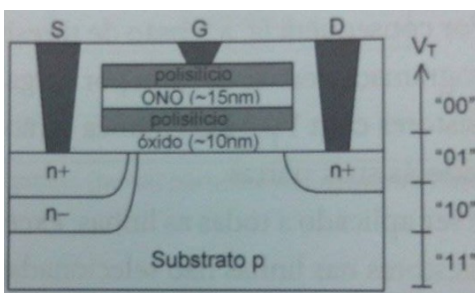


Figura 7: Célula Multibit (MCL)

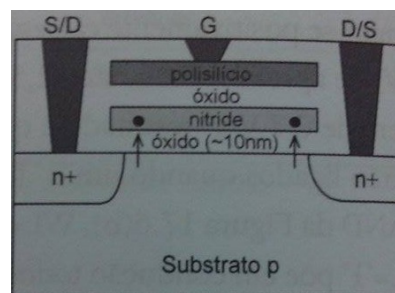


Figura 8: Célula Multibit (MBC)

Ambas as células, MCL E MBC, funcionam com tunelamento para apagamento e injeção por avalanche (*CHEI*) ou tunelamento para programação.

2.6 Utilização da Memória Flash em FPGA

No interior de cada bloco lógico de FPGA existem vários modos possíveis para implementação de funções lógicas. O mais utilizado é o bloco de memória LUT (*Look-Up Table*). As células de armazenamento de um LUT são voláteis, o que implicada a perda de conteúdo, no caso de falta de energia elétrica. Dessa forma, o FPGA deve ser programado toda vez que for energizado. Para evitar a perda da programação, utiliza-se uma pequena memória flash, em o conteúdo da LUT é programado nas células ETOX da memória flash, cuja função é carregar as células de armazenamento, toda vez que o FPGA for energizado.

2.7 Utilização da Memória Flash em CPLD

Nos CPLDs em cada célula lógica, também conhecida como *Logic Array Block* (LAB), é constituída por unidades básicas conhecidas com *Logic Elements* (LE). Os *Logic Elements* por sua vez possuem *LUTs*, *Flip-Flops* (implementam registradores) e lógicas para interconexão entre *Logic Elements* adjacentes (*Carry Chain* e *Cascade Chain*).

A programação tanto da parte lógica quanto a parte da configuração das interconexões nos CPLDs é toda armazenada em células flash, explicadas anteriormente. Diferente do que ocorre no FPGA que essa configuração é toda armazenada na memória SRAM e que, em alguns casos, utiliza-se memória flash para salvar os dados da memória SRAM.

Conclusão

Este trabalho apresentou dois tipos de memórias utilizadas para programar CPLDs e FPGAs. No FPGA, a memória SRAM controla os nós das portas dos transistores de passagem e as linhas de seleção do multiplexador que alimenta as entradas dos blocos lógicos; e a memória flash é utilizada nos FPGAs para evitar a perda da programação em caso de falta de energia elétrica, devido à volatilidade das células de armazenamento da LUT. Já no CPLD, as células flash armazenam a programação da parte lógica e da parte da configuração das interconexões.

Referências

Volnei A. Pedroni. Eletrônica Digital Moderna e Vhdl. Elsevier/Campus: 2010.

Ferreira, Elnatan Chagas. Circuitos Lógicos Programáveis. Disponível em: <http://www.demic.fee.unicamp.br/el-natan/ee610>. Acesso em: 20 fev. 2015.

Filho, Eduardo F. de Simas. Dispositivos de Hardware Programável. Disponível em: <http://www.dee.eng.ufba.br/hom> 20 fev. 2015.

Zaghetto, Alexandre; Prado, Augusto Cesar; Tavares, André. Hcpld - High Capacity Programable Logic Devices. Disponível em: http://www.gta.ufrj.br/grad/01_1/pld/hcpld.htm. Acesso em : 20 fev. 2015.