Capítulo 3-Circuitos Sequenciais

Blocos padrão sequenciais

Profa. Eliete Caldeira

Blocos padrão sequenciais

- Existem circuitos sequenciais que por serem muito utilizados, são disponibilizados como dispositivos de prateleira
- Entre estes dispositivos estão registradores, registradores de deslocamento, contadores, incrementadores, temporizadores,

Registrador de N bits

- É um componente sequencial capaz de armazenar N bits.
- Larguras típicas: 8, 16 e 32, embora qualquer largura seja possível.

- Ações básicas:
  - Carregar, escrever ou armazenar dados em um registrador
  - Ler um registrador, que consiste em se conectar as saídas do registrador
- A leitura não está sincronizada com o relógio
- Além disto, a leitura não remove os bits do registrador nem os modifica de nenhum modo

Registrador de N bits

- Tipo mais básico de registrador
  - Conjunto de flip-flops que são carregados a cada ciclo de relógio.
- Deve ser carregado em todos os ciclos de relógio

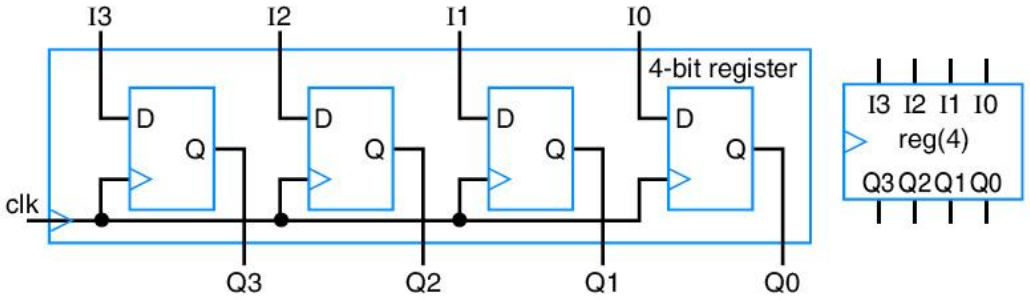


Figure 3.30 A basic 4-bit register internal design (left) and block symbol (right).

- Útil como registrador de estado em um bloco de controle

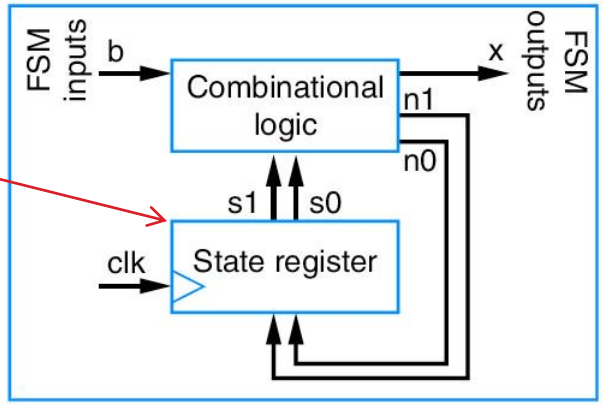
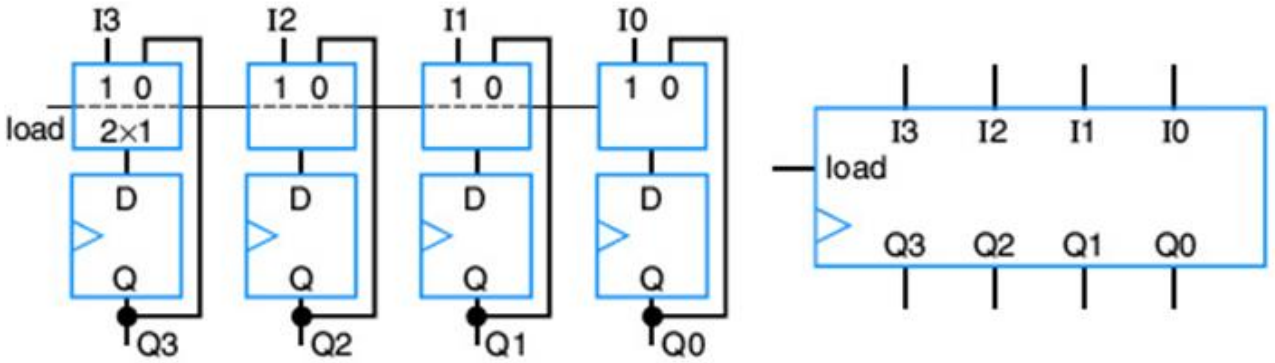


Figure 3.47 Standard controller architecture for the laser timer.

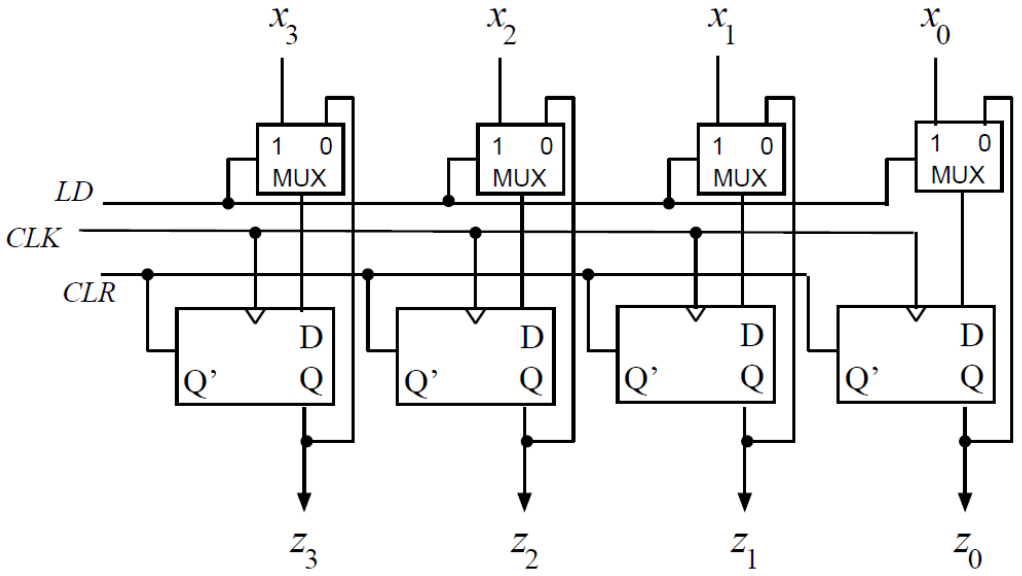
Registrador de carga paralela

- Pode-se obter controle sobre a carga de um registrador colocando um multiplexador 2x1.
- Na borda de subida do clock:
  - Se load=0 cada flip-flop será carregado com o valor de sua própria saída Q
  - Se load=1 cada flip-flop será carregado com uma das entradas de dados



Registrador de carga paralela

- No circuito da figura, CLR é uma entrada de reset assíncrona e LD é a entrada que controla sincronamente a carga
  - Se LD = 0 mantém a memória
  - Se LD = 1 carrega novo valor



Registrador de carga paralela

- Exemplo:Para o circuito, as entradas são como na figura da direita

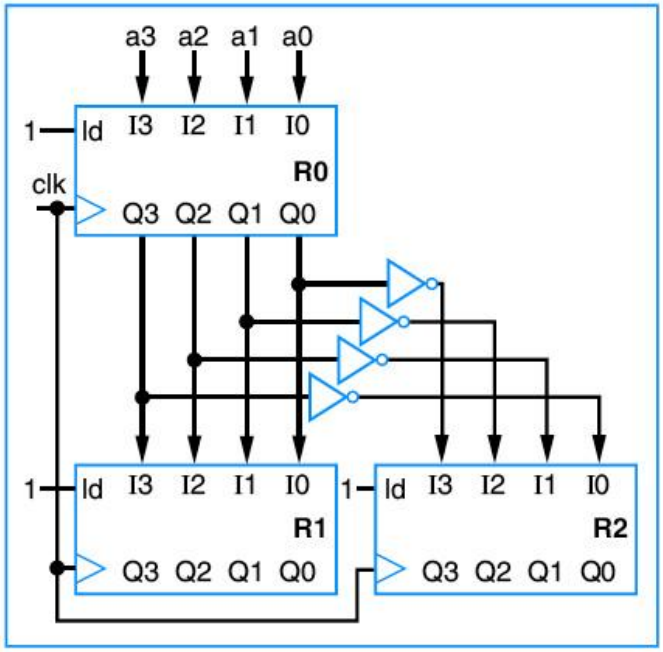


Figure 4.2 Basic register example.

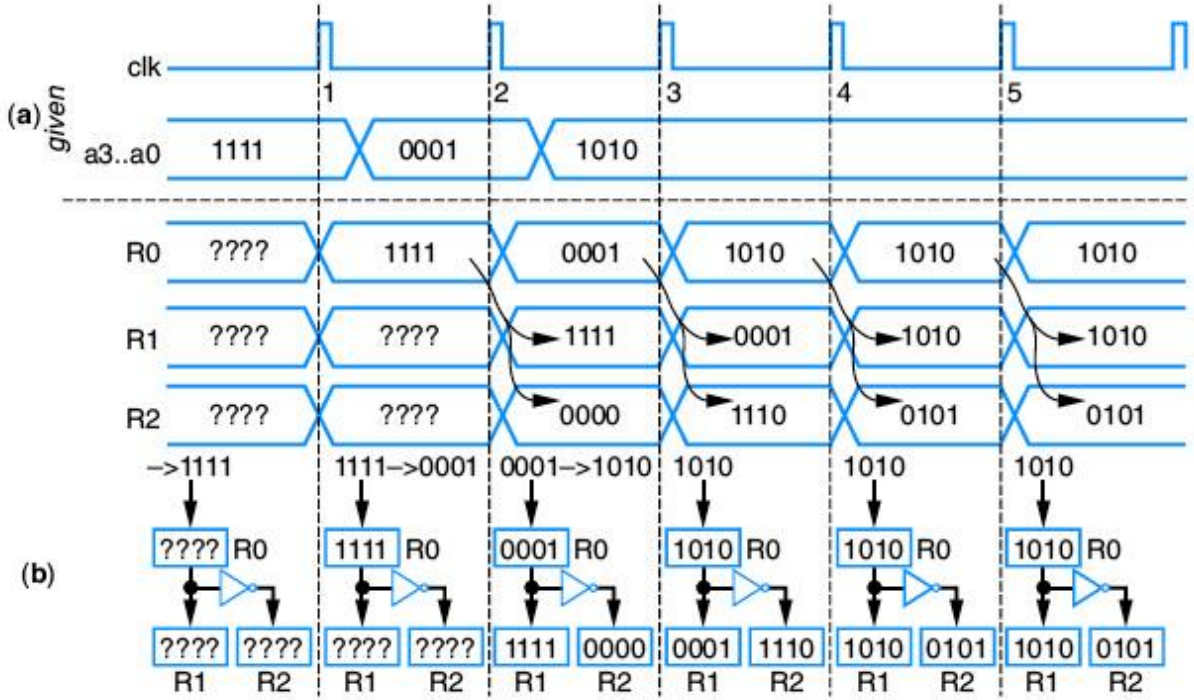


Figure 4.3 Basic register example: (a) timing diagram, and (b) the contents of each register.

Registrador de carga paralela

- Exemplo: Balança com display que mostra peso atual e peso salvo anteriormente
  - Pode ser usada para comparar pesos
  - Um botão salva o valor atual da balança

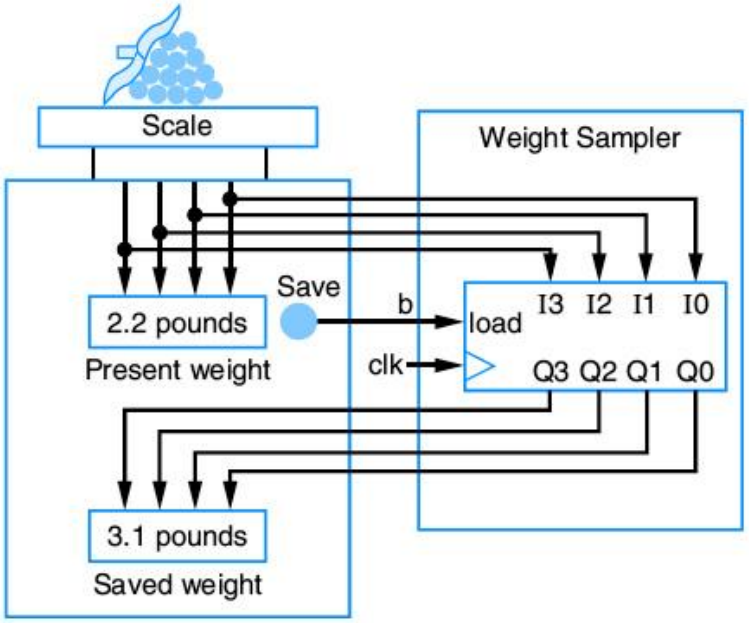


Figure 4.4 Weight sampler implemented using a 4-bit parallel load register.

Registrador de deslocamento

- Desloca os conteúdos de um registrador para a esquerda ou para a direita

Registrador de deslocamento

- Na figura o deslocamento é para a direita

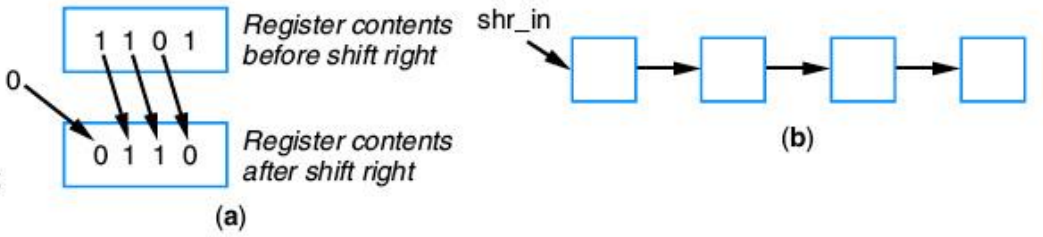


Figure 4.10 Right shift example: (a) sample contents before and after a right shift, and (b) bit-by-bit view of the shift.

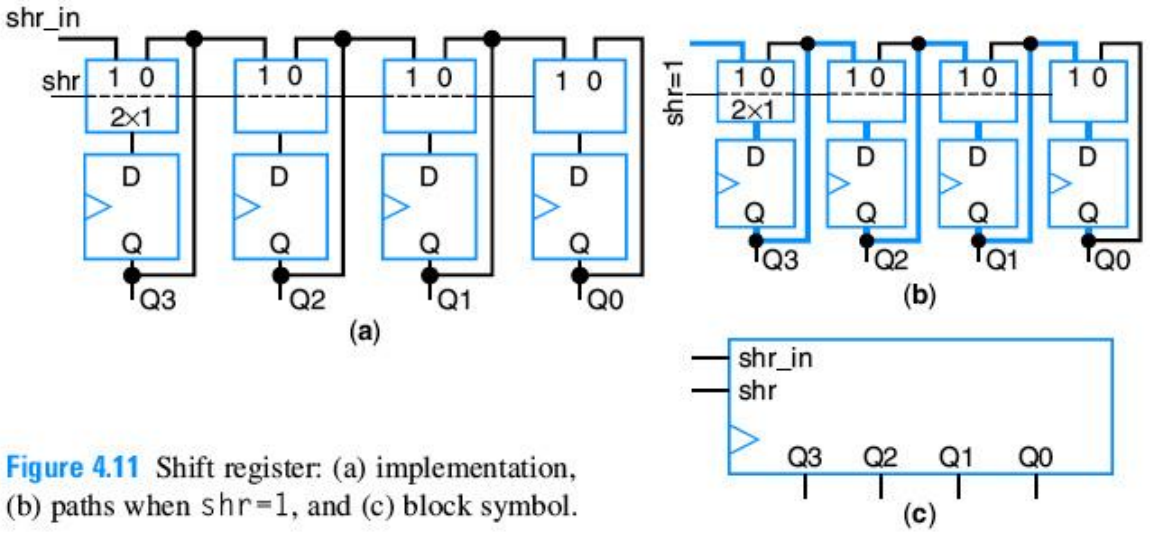


Figure 4.11 Shift register: (a) implementation, (b) paths when `shr=1`, and (c) block symbol.

shr = 0 → mantém memória
shr = 1 → desloca para a direita
shr_in → bit que será inserido

Registrador circular

- Variação do registrador de deslocamento
- Para o deslocamento circular para a direita

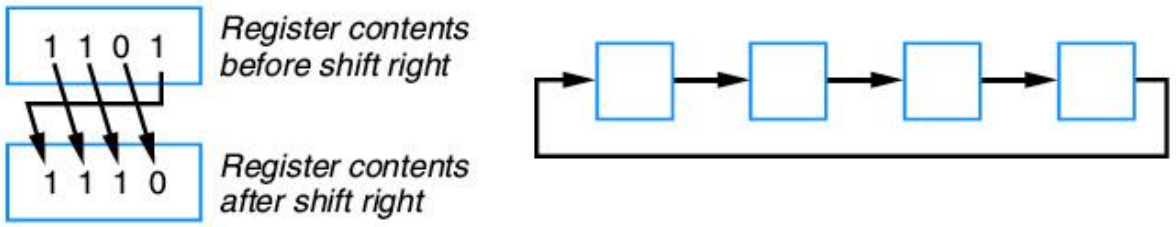


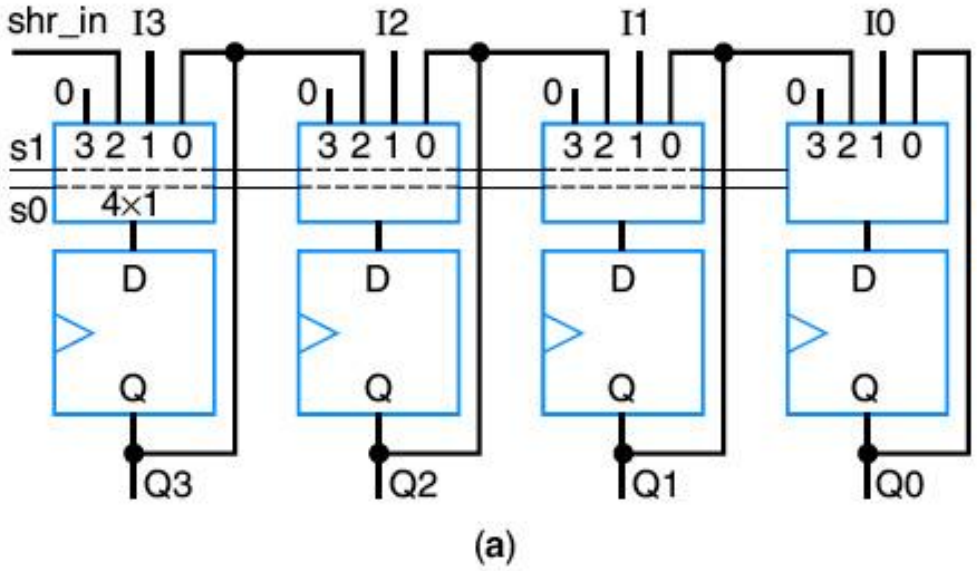
Figure 4.12 Right rotate example: (a) register contents before and after the rotate, and (b) bit-by-bit view of the rotate operation.

Registradores com múltiplas funções

- Realizam uma variedade de operações (funções):
  - Carga, deslocamento à direita, deslocamento à esquerda, rotação à direita, rotação à esquerda, etc.
- O usuário do registrador seleciona a operação desejada definindo as entradas de controle

Registrador com carga paralela e deslocamento à direita

- O usuário seleciona a função com s1 e s0



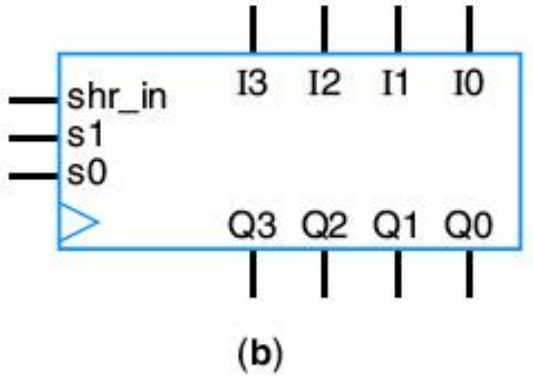


Figure 4.14 4-bit register with parallel load and shift right operations: (a) internal design, and (b) block symbol.

s1	s0	Operation
0	0	Maintain present value
0	1	Parallel load
1	0	Shift right
1	1	(unused – let's load 0s)

Figure 4.15 Operation table of a 4-bit register with parallel load and shift right operations.

Registrador com carga paralela e deslocam. à direita e à esquerda

- O usuário seleciona a função com s1 e s0

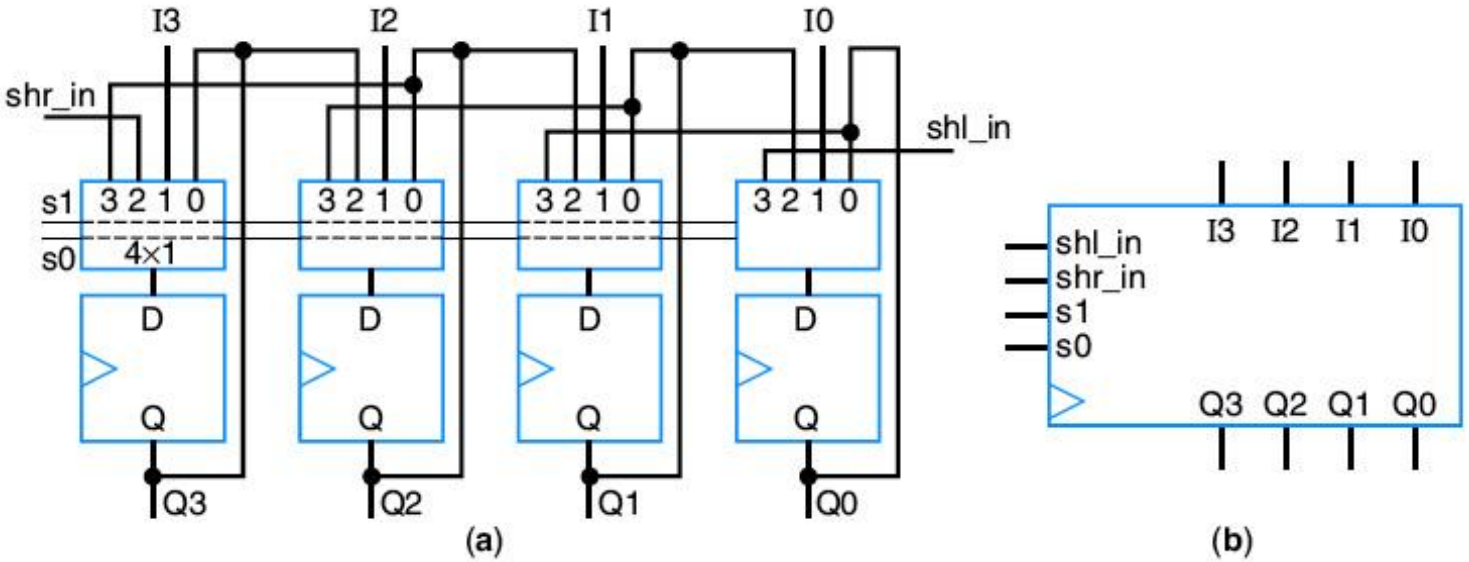


Figure 4.16 4-bit register with parallel load, shift left, and shift right operations: (a) internal design, (b) block symbol.

s1	s0	Operation
0	0	Maintain present value
0	1	Parallel load
1	0	Shift right
1	1	Shift left

Figure 4.17 Operation table of a 4-bit register with parallel load, shift left, and shift right operations.

Registrador com carga paralela e deslocam. à direita e à esquerda

- Geralmente, os registradores não vêm com entradas de controle que codificam a função usando o número mínimo de bits
- Cada função tem a própria entrada de controle

Inputs			Outputs		Note
ld	shr	shl	s1	s0	Operation
0	0	0	0	0	Maintain value
0	0	1	1	1	Shift left
0	1	0	1	0	Shift right
0	1	1	1	0	Shift right
1	0	0	0	1	Parallel load
1	0	1	0	1	Parallel load
1	1	0	0	1	Parallel load
1	1	1	0	1	Parallel load

(a)

ld	shr	shl	Operation
0	0	0	Maintain value
0	0	1	Shift left
0	1	X	Shift right
1	X	X	Parallel load

(b)

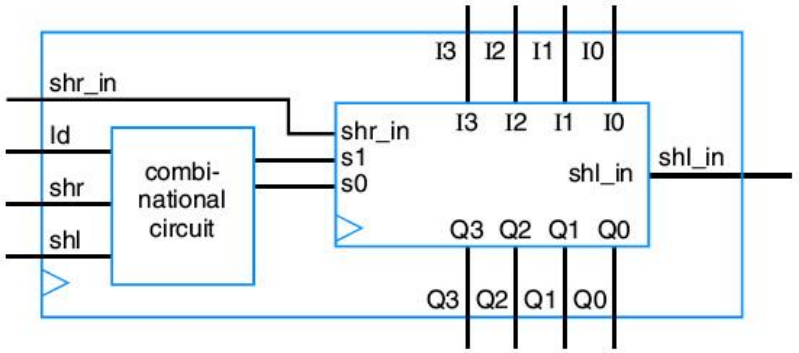


Figure 4.19 A small combinational circuit maps the control inputs `ld`, `shr`, and `shl` to the mux select inputs `s1` and `s0`.

Registrador com carga paralela e deslocam. à direita e à esquerda

- Geralmente, os registradores não vêm com entradas de controle que codificam a função usando o número mínimo de bits
- Cada função tem a própria entrada de controle

Inputs			Outputs		Note
ld	shr	shl	s1	s0	Operation
0	0	0	0	0	Maintain value
0	0	1	1	1	Shift left
0	1	0	1	0	Shift right
0	1	1	1	0	Shift right
1	0	0	0	1	Parallel load
1	0	1	0	1	Parallel load
1	1	0	0	1	Parallel load
1	1	1	0	1	Parallel load

(a)

ld	shr	shl	Operation
0	0	0	Maintain value
0	0	1	Shift left
0	1	X	Shift right
1	X	X	Parallel load

(b)

E preciso fazer este circuito combinacional que converte os sinais separados no código de controle

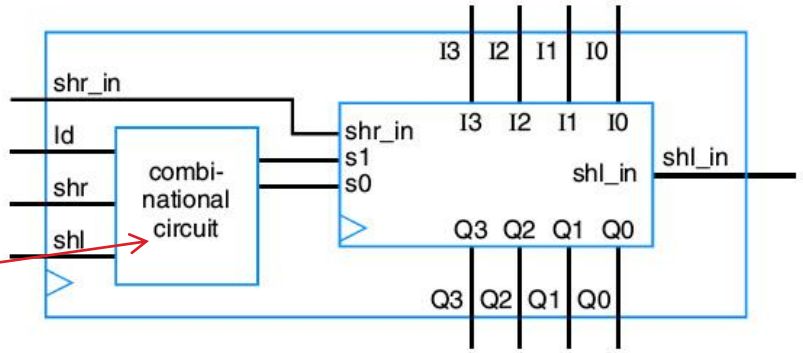


Figure 4.19 A small combinational circuit maps the control inputs `ld`, `shr`, and `shl` to the mux select inputs `s1` and `s0`.

Processo de projeto de registradores

Passos	Explicação
1. Determine o tamanho do multiplexador	Conte o número de funções e coloque um multiplexador à frente de cada flip-flop com, no mínimo, esse número de entradas de dados.
2. Crie a tabela de funções do multiplexador	Crie uma tabela de funções que define as operações desejadas para cada valor possível das linhas de seleção do multiplexador.
3. Conecte as entradas do multiplexador	Para cada função, conecte a entrada de dados correspondente do multiplexador à entrada externa ou saída de flip-flop apropriada para obter a função desejada.
4. Mapeie as linhas de controle	Crie uma tabela-verdade que mapeia as linhas de controle externas nas linhas internas de seleção dos multiplexadores, com prioridades apropriadas, e então projete a lógica combinacional que implemente esse mapeamento

Processo de projeto de registradores

- Exemplo: Projete um registrador com carga, deslocamento, *set* e *clear* síncronos

Processo de projeto de registradores

- Exemplo: Projete um registrador com carga, deslocamento à esquerda, *set* e *clear* síncronos
- Passo 1: Determine o tamanho do multiplexador
  - São cinco funções: carga, deslocamento à esquerda, *clear* síncrono, *set* síncrono e manutenção do valor atual
- Passo 2: Crie a tabela de funções do multiplexador

Processo de projeto de registradores

- Exemplo: Projete um registrador com carga, deslocamento à esquerda, *set* e *clear* síncronos
- Passo 2: Crie a tabela de funções do multiplexador

s2	s1	s0	Operation
0	0	0	Maintain present value
0	0	1	Parallel load
0	1	0	Shift left
0	1	1	Synchronous clear
1	0	0	Synchronous set
1	0	1	Maintain present value
1	1	0	Maintain present value
1	1	1	Maintain present value

Figure 4.21 Operation table for a register with load, shift, and synchronous clear and set.

Processo de projeto de registradores

- Exemplo: Projete um registrador com carga, deslocamento à esquerda, *set* e *clear* síncronos
- Passo 3: Conecte as entradas do multiplexador

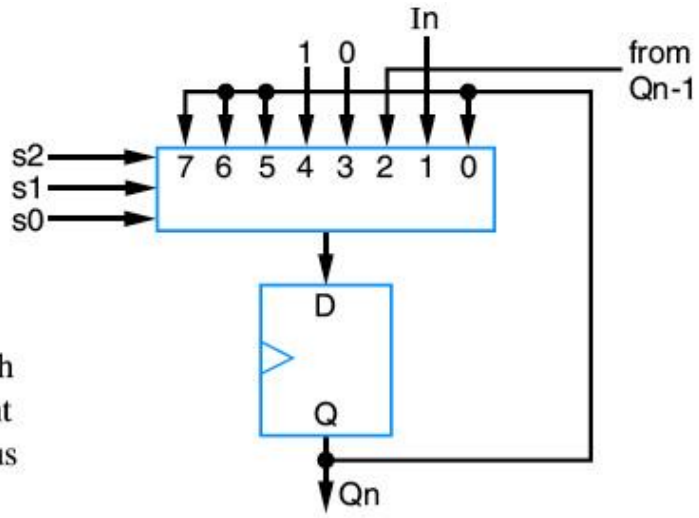


Figure 4.22 *N*th bit-slice of a register with the following operations: maintain present value, parallel load, shift left, synchronous clear, and synchronous set.

Processo de projeto de registradores

- Exemplo: Projete um registrador com carga, deslocamento à esquerda, *set* e *clear* síncronos
- Passo 4: Mapeie as linhas de controle

Inputs				Outputs			
clr	set	ld	shl	s2	s1	s0	Operation
0	0	0	0	0	0	0	Maintain present value
0	0	0	1	0	1	0	Shift left
0	0	1	X	0	0	1	Parallel load
0	1	X	X	1	0	0	Set to all 1s
1	X	X	X	0	1	1	Clear to all 0s

Figure 4.23 Truth table for the control lines of a register with the *N*th bit-slice shown in Figure 4.22.

Processo de projeto de registradores

- Exemplo: Projete um registrador com carga, deslocamento à esquerda, *set* e *clear* síncronos
- Passo 4: Mapeie as linhas de controle

Inputs				Outputs			
clr	set	ld	shl	s2	s1	s0	Operation
0	0	0	0	0	0	0	Maintain present value
0	0	0	1	0	1	0	Shift left
0	0	1	X	0	0	1	Parallel load
0	1	X	X	1	0	0	Set to all 1s
1	X	X	X	0	1	1	Clear to all 0s

Figure 4.23 Truth table for the control lines of a register with the *N*th bit-slice shown in Figure 4.22.

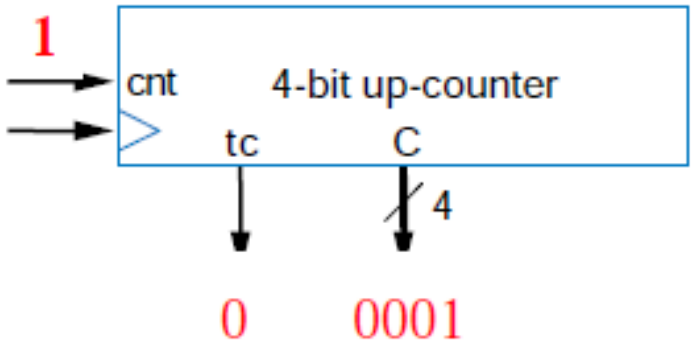
```
s2 = clr'*set
s1 = clr'*set'*ld'*shl + clr
s0 = clr'*set'*ld + clr
```

Contador de N bits

- Componente construído a partir de uma extensão de um registrador de N bits:
  - Pode incrementar ou decrementar o próprio valor a cada ciclo de relógio
  - Pode ter uma entrada de habilitação para manter o valor (cnt = 0) ou contar (cnt = 1)
  - Incrementar significa adicionar 1, decrementar significa subtrair 1
- Contador que pode incrementar: contador crescente (*up-counter*)
- Contador que pode decrementar: contador decrescente (*down-counter*)
- Contador que pode incrementar e decrementar: contador crescente/decrescente (*up-down-counter*)
- Contagem terminal (ou *terminal counter*) = 1 durante o ciclo de relógio no qual o contador atinge seu último valor de contagem

Contador de N bits

- Na figura o contador crescente de 4 bits mantém a contagem se cnt = 0 e incrementa se cnt = 1
- Na situação mostrada, as saídas após a borda de subida do clock serão C=0010 e tc = 0



Contador crescente

- Projeto de um contador crescente de N bits usando o processo de projeto de registradores descrito anteriormente
  - Valor incrementado do registrador irá alimentar uma entrada de dados do MUX
  - Linhas de controle do contador serão mapeadas para linhas de seleção do MUX

Contador crescente

- Projeto de um contador crescente de N bits em uma visão mais simples:
  - Registrador
  - Componente incrementador para somar 1
  - cnt=0, o registrador deve manter o seu valor corrente.
  - cnt=1, o registrador deve ser carregado com o seu valor corrente + 1

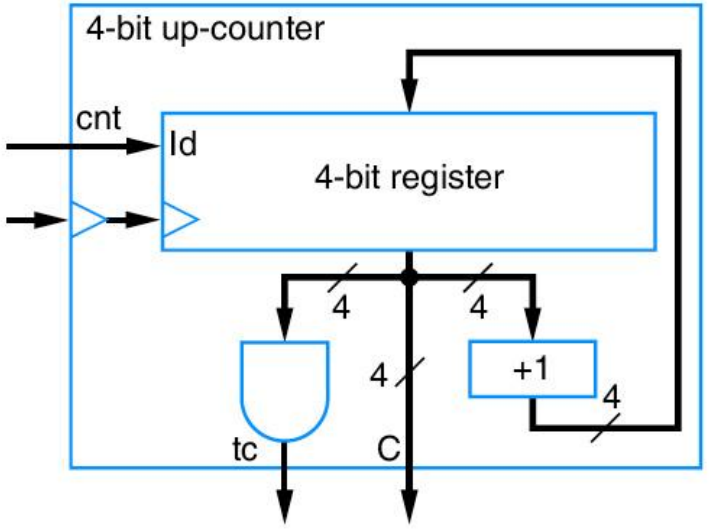


Figure 4.48 4-bit up-counter internal design.

Contador crescente

- Incrementador: Pode-se usar um somador de N bits, colocando na entrada B o valor 0001 e um 0 na entrada de "vem um".
- Mas não é preciso toda a lógica envolvida em somador de N bits, uma vez que B sempre será 0001.
- Somar 1 a um número binário envolve apenas dois bits por coluna, e não três. Assim, pode-se usar meios-somadores

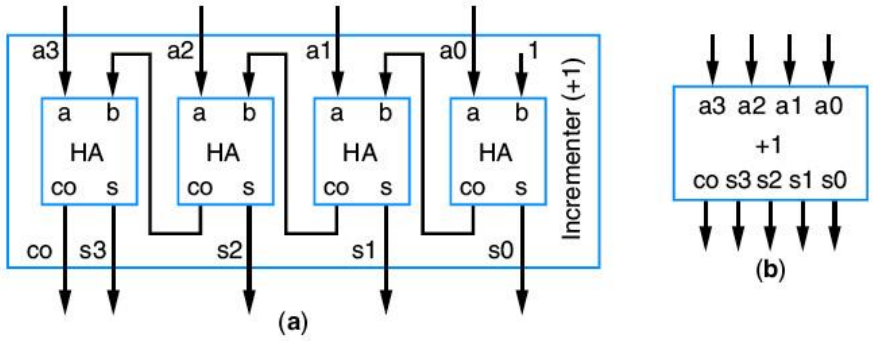


Figure 4.50 4-bit incrementer: (a) internal design, and (b) block symbol.

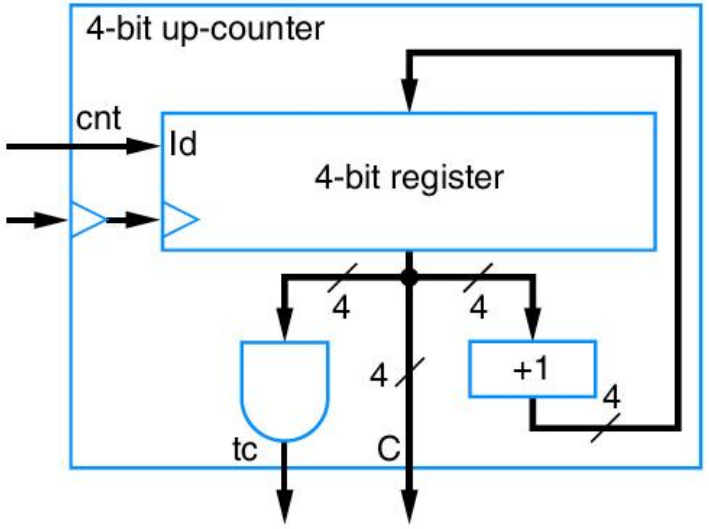


Figure 4.48 4-bit up-counter internal design.

Contador crescente

- Podemos projetar um incrementador usando o processo de projeto lógico combinacional
  - Vantagem: Atraso total de apenas dois níveis de portas
- Incrementadores de mais bits a construção de somadores usando o processo de projeto lógico combinacional não é muito prática

Inputs				Outputs				
a3	a2	a1	a0	c0	s3	s2	s1	s0
0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	1	1
0	0	1	1	0	0	1	0	0
0	1	0	0	0	0	1	0	1
0	1	0	1	0	0	1	1	0
0	1	1	0	0	0	1	1	1
0	1	1	1	0	1	0	0	0
1	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	1	0
1	0	1	0	0	1	0	1	1
1	0	1	1	0	1	1	0	0
1	1	0	0	0	1	1	0	1
1	1	0	1	0	1	1	1	0
1	1	1	0	0	1	1	1	1
1	1	1	1	1	0	0	0	0

Figure 4.51 Truth table for four-bit incrementer.

Contador crescente

- Exemplo: Gerador de pulsos de 1Hz usando um oscilador de 256Hz

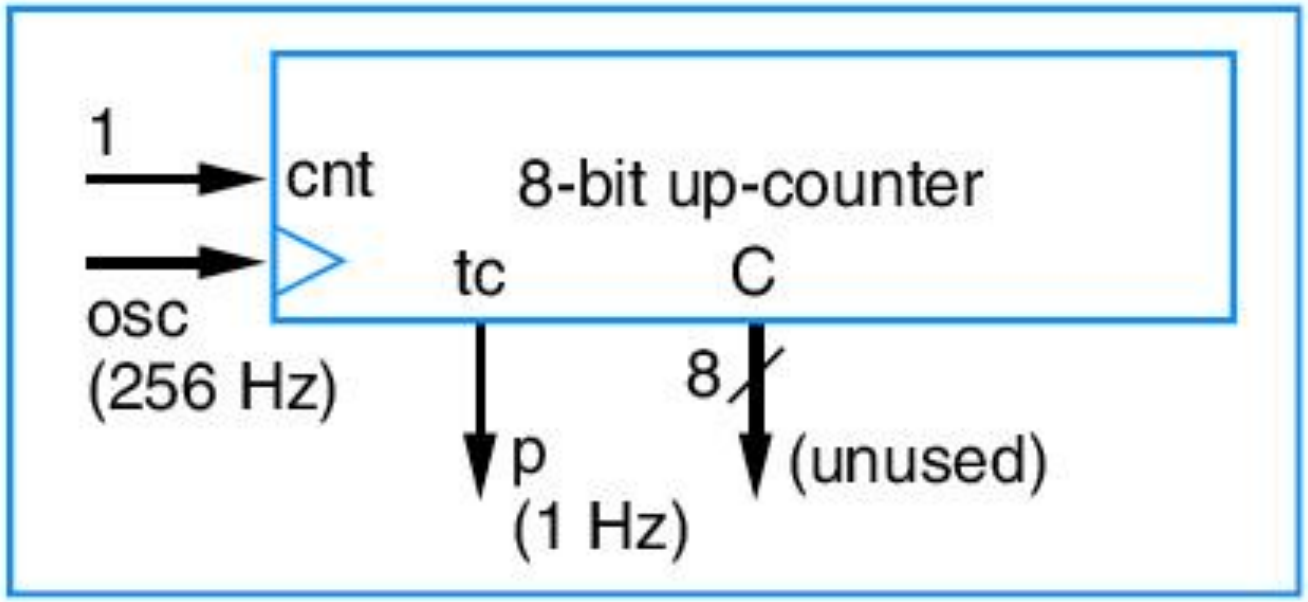


Figure 4.53 Clock divider.

Contador decrescente

- Projeto de um contador decrescente de N bits em uma visão mais simples:
  - Registrador
  - Componente decrementador para subtrair 1
  - cnt=0, o registrador deve manter o seu valor corrente.
  - cnt=1, o registrador deve ser carregado com o seu valor corrente -1
- Decrementador: pode ser feito invertendo-se os bits de C, somando 1 e invertendo novamente. Ex: 0100 ->1011+1=1100->0011

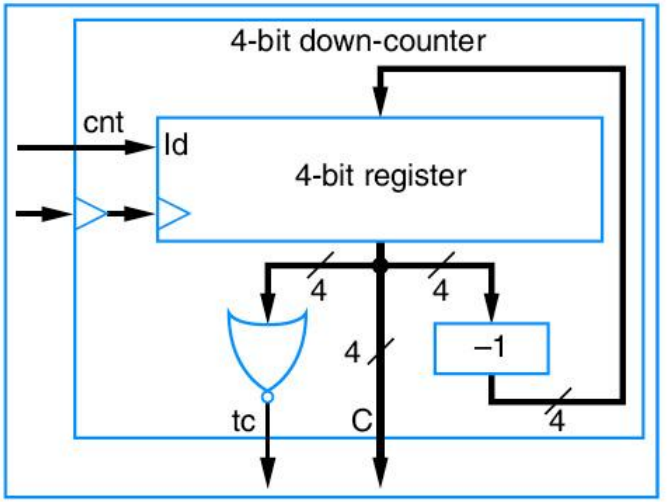


Figure 4.54 4-bit down-counter design.

Contador crescente/decrescente

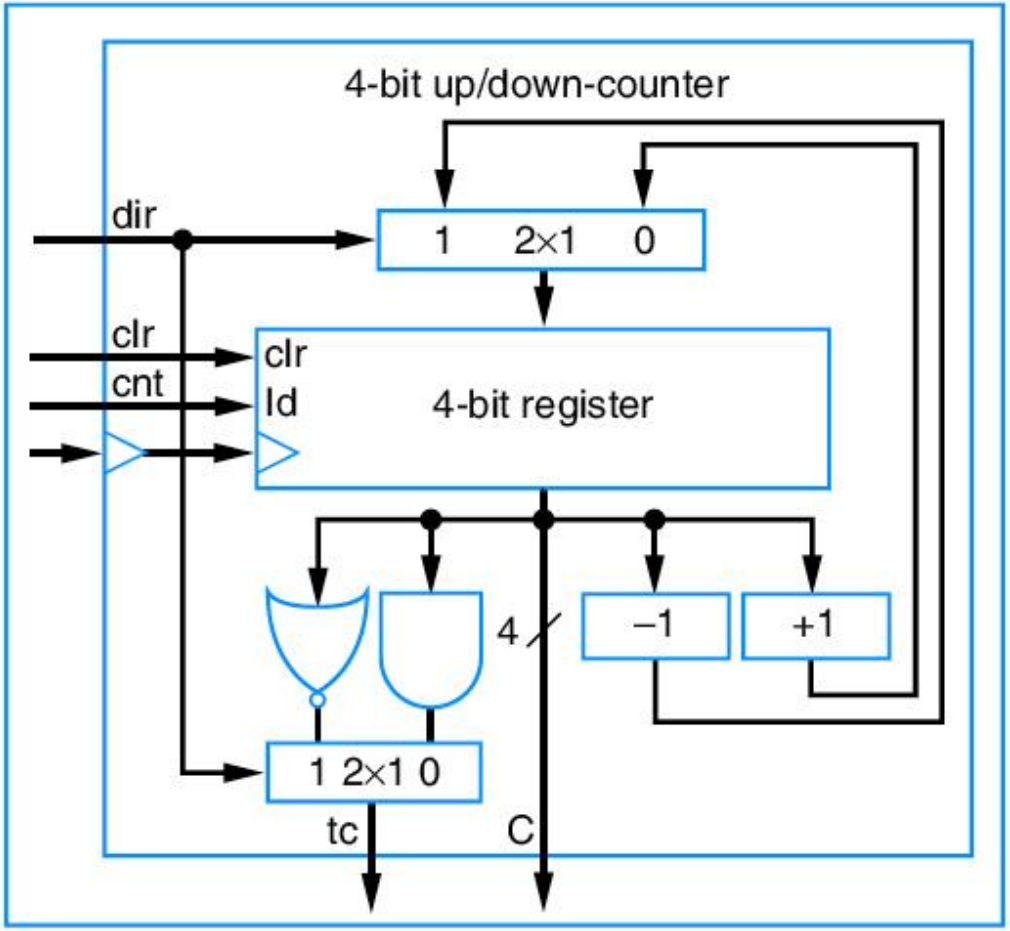


Figure 4.55 4-bit up/down-counter design.

Contador com carga paralela

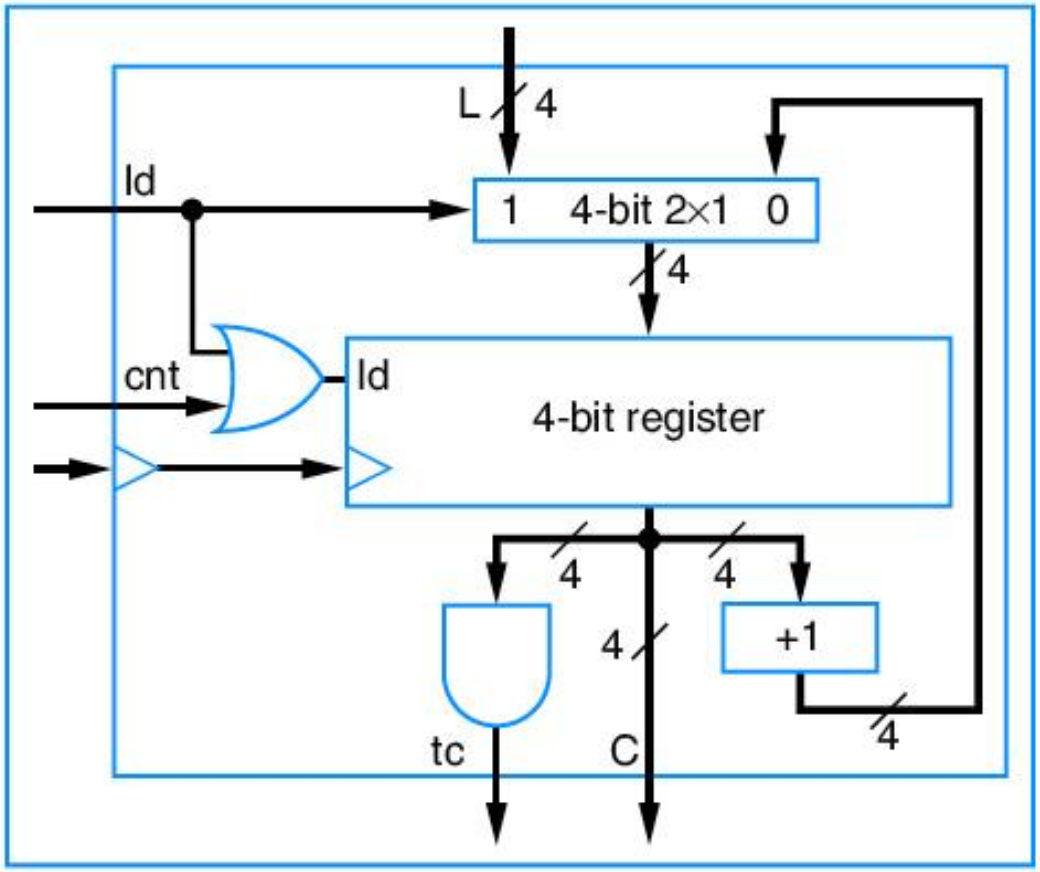
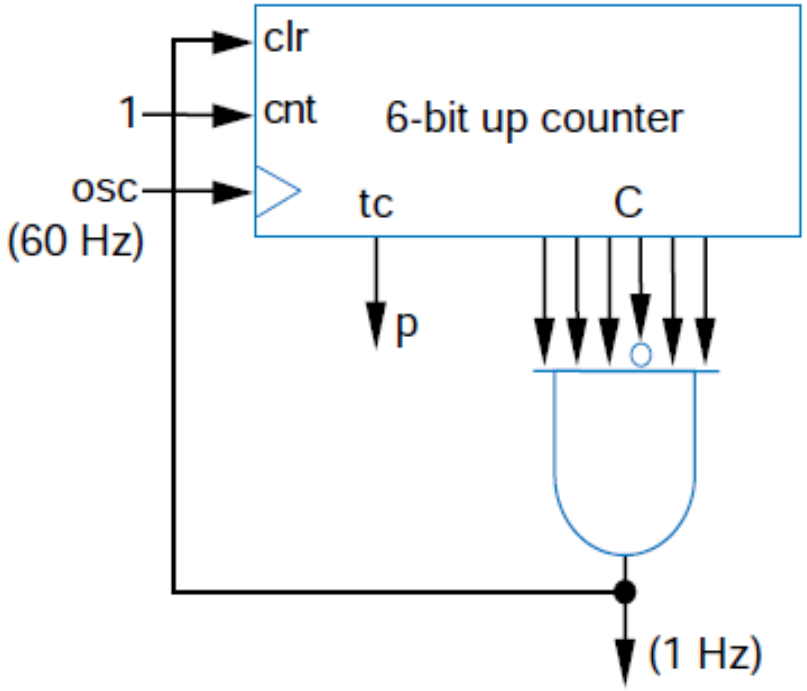


Figure 4.57 Internal design of a 4-bit up-counter with load.

Contadores

- Exemplo: Gerador de pulsos de 1 Hz usando um oscilador de 60 Hz
- Contador de 6 bits:
  - Conta de 0 a 63
  - Lógica para indicar 59 e
  - Resetar o contador



Temporizador

- Tipo especial de contador que mede tempo
- A medição de tempo é uma tarefa muito comum em sistemas digitais
- Se soubermos a duração de um ciclo de relógio, multiplicaremos o número de ciclos pela duração de um ciclo de relógio para obter a duração total do evento

Temporizador

- Exemplo: Sistema de medição de velocidade de rodovia

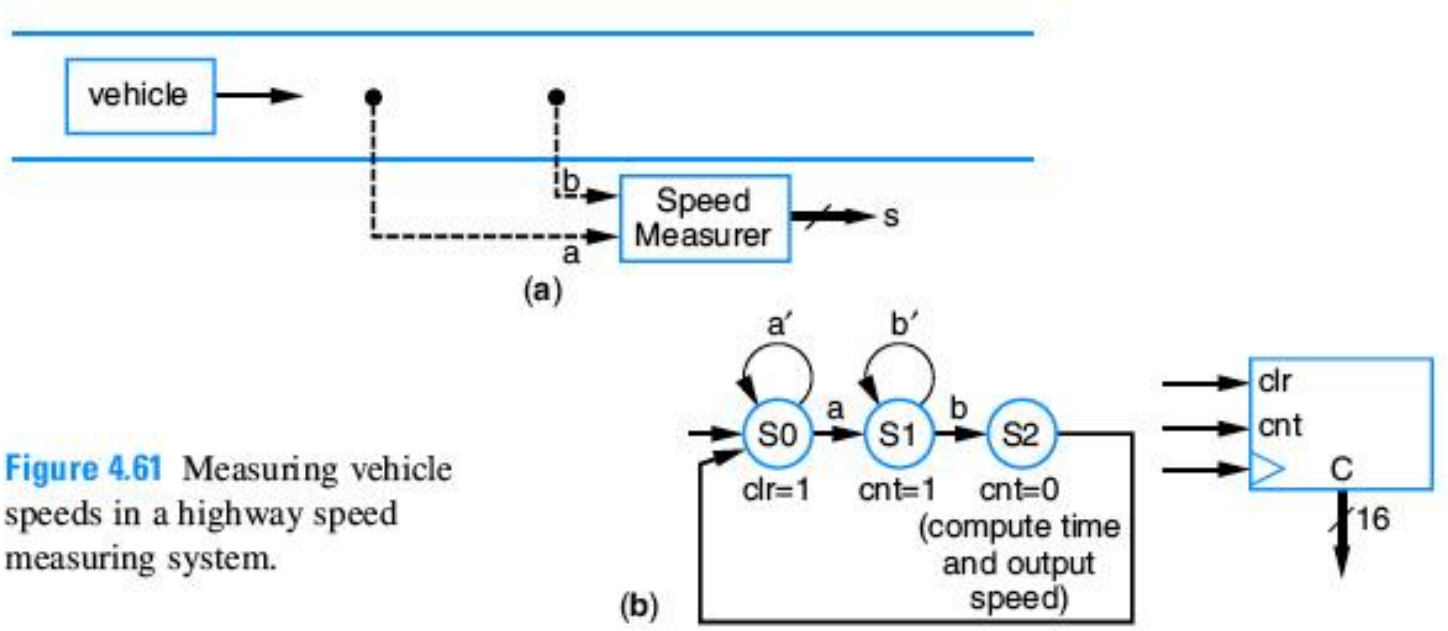


Figure 4.61 Measuring vehicle speeds in a highway speed measuring system.

Temporizador

- Exemplo: Sistema de medição de velocidade de rodovia
  - Sensor a reseta e inicia o temporizador
  - Sensor b para o temporizador
- Conhecendo-se a frequência do temporizador e a distancia entre sensores, calcula-se a velocidade

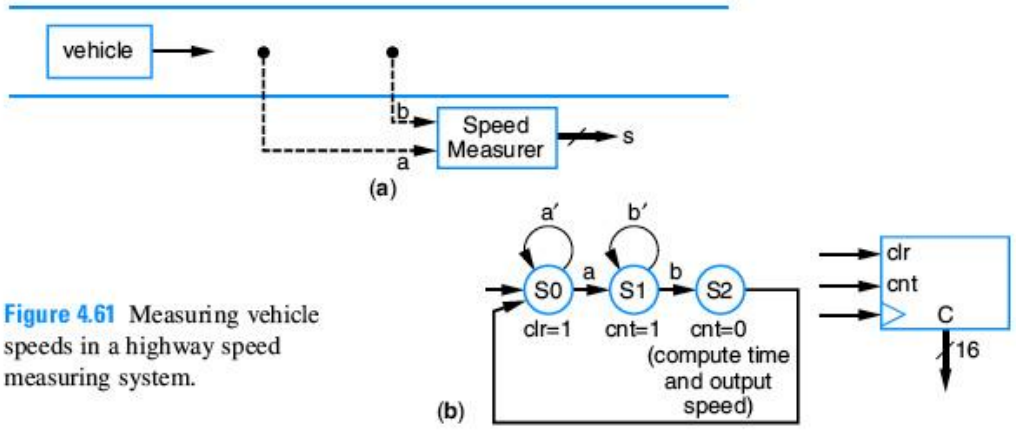


Figure 4.61 Measuring vehicle speeds in a highway speed measuring system.

Banco de Registradores MxN

- Componente de memória de blocos operacionais
- Propicia um acesso eficiente a um conjunto de M registradores
- Cada registrador tem uma largura de N bits
- Um registrador ativo por vez

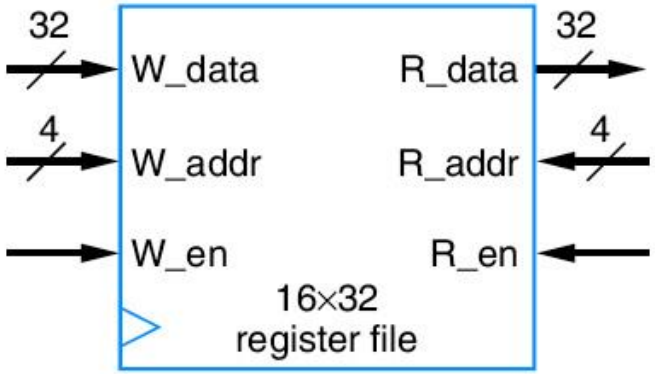


Figure 4.80 16x32 register file block symbol.

Banco de Registradores MxN

- Usar para evitar:
  - Acúmulo de muitos fios, congestionamento
  - Ramificações de um fio, fanout
  - Correntes nos fios das ramificações sejam pequenas demais para poder controlar eficientemente os transistores
- Usar quando:
  - Não se necessita carregar mais de um registrador de cada vez
  - Não precisamos ler mais de um registrador de cada vez.
- Um banco de registradores MxN resolve os problemas de fanout e congestionamento
- M registradores são agrupados em um único componente:
  - Entrada e Saida de dados de N bits de largura

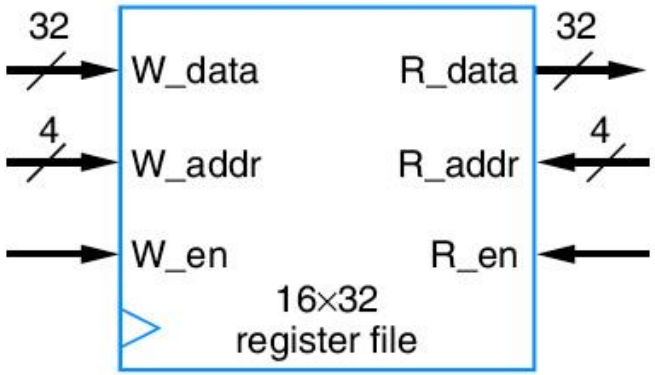


Figure 4.80 16x32 register file block symbol.

Banco de Registradores MxN

- Implementação

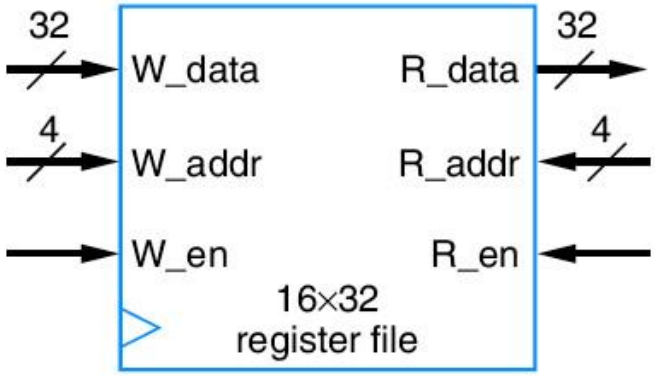


Figure 4.80 16x32 register file block symbol.

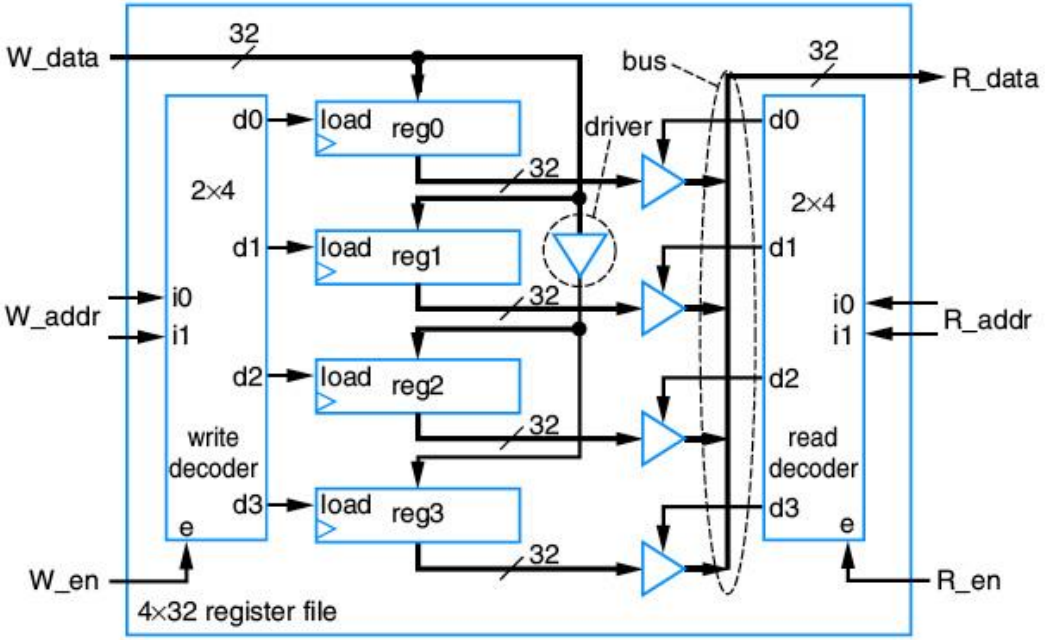
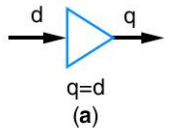


Figure 4.81 One possible internal design of a 4x32 register file.



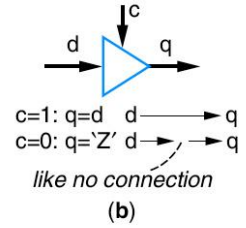


Figure 4.82 (a) driver, (b) three-state driver.

Banco de Registradores MxN

- Escrita
- Coloca-se:
  - Dados a serem escritos na entrada W_data
  - Endereço do registrador na entrada W_addr
  - Indicar que se deseja escrever (W_en = 1)

- **W_data + W_addr + W_en = Escrita**

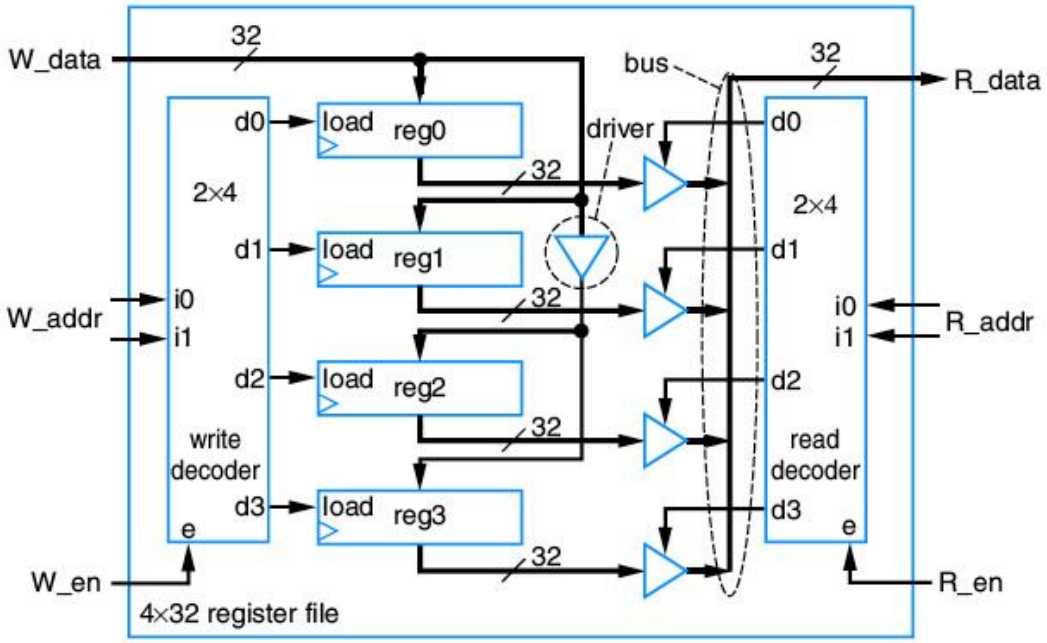


Figure 4.81 One possible internal design of a 4x32 register file.

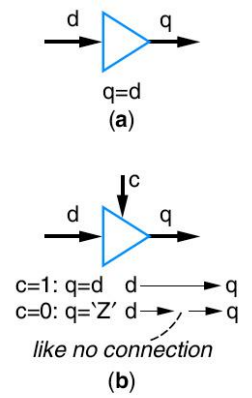


Figure 4.82 (a) driver, (b) three-state driver.

Banco de Registradores MxN

- Leitura
- Coloca-se:
  - Endereço do registrador a ser lido, R_addr
  - Habilita-se a leitura, R_en=1
  - Banco de registradores coloca na saída R_data o conteúdo do registrador que foi endereçado

- **R_data + R_addr + R_en = Leitura**

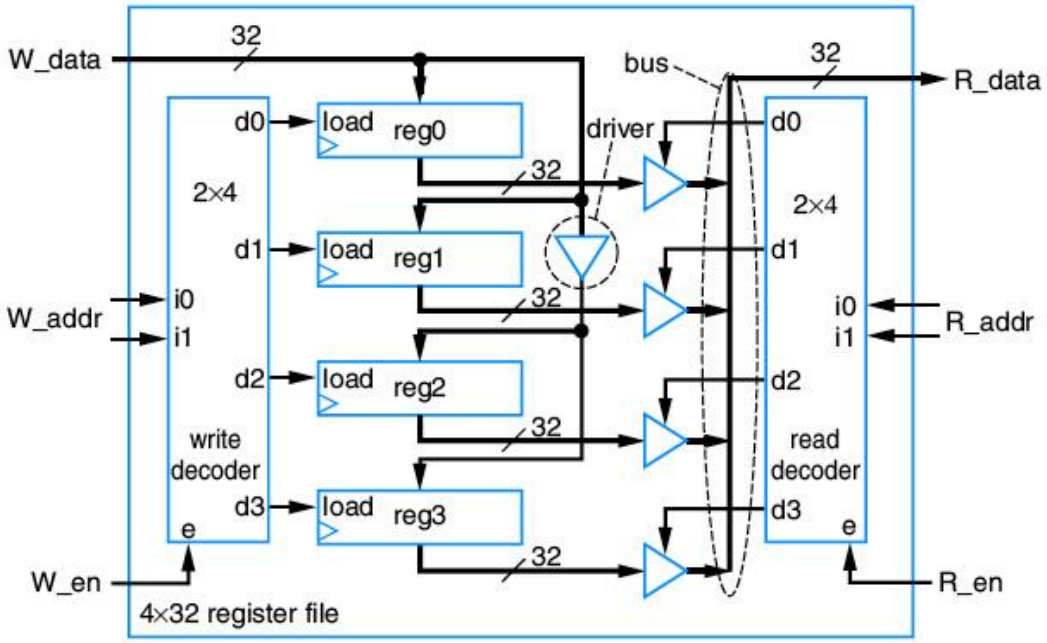


Figure 4.81 One possible internal design of a 4x32 register file.

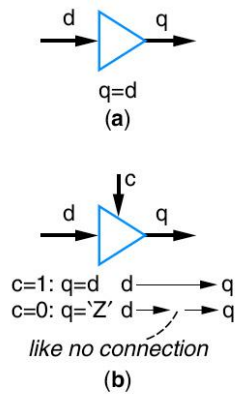


Figure 4.82 (a) driver, (b) three-state driver.

Banco de Registradores MxN

- *Driver* ou *Buffer*
  - Saída de um *driver* é igual à sua entrada
  - Sinal de saída é mais robusto (corrente mais elevada)
  - Solucao para o fanout
- *Driver* de três estados ou *Buffer* de três estados.
  - c = 1, o componente atua como um *driver* comum
  - c = 0, a saída no estado de alta impedância (‘Z’). É como se não houvesse nenhuma conexão entre a entrada e a saída do *driver*

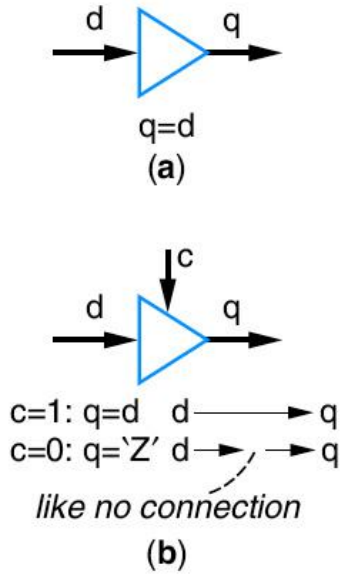


Figure 4.82 (a) driver, (b) three-state driver.

Banco de Registradores MxN

- Escrita e Leitura
- A escrita e leitura são independentes entre si.
- Durante o mesmo ciclo de relógio, pode-se:
  - Escrever em um registrador e
  - Ler de outro (ou do mesmo) registrador

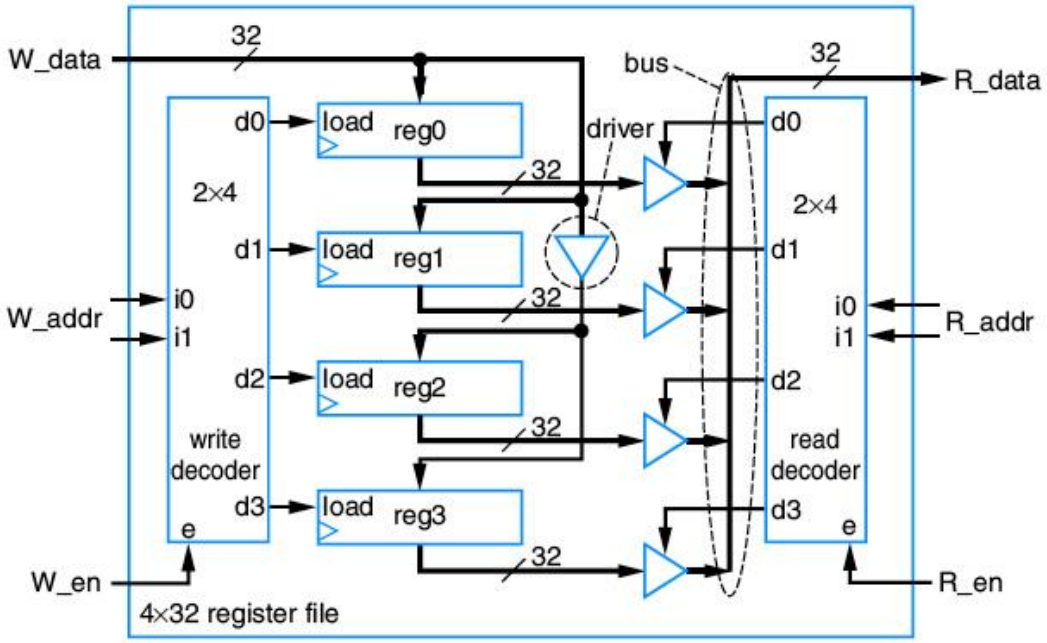


Figure 4.81 One possible internal design of a 4x32 register file.

Banco de Registradores MxN

- Lendo e escrevendo ao mesmo tempo

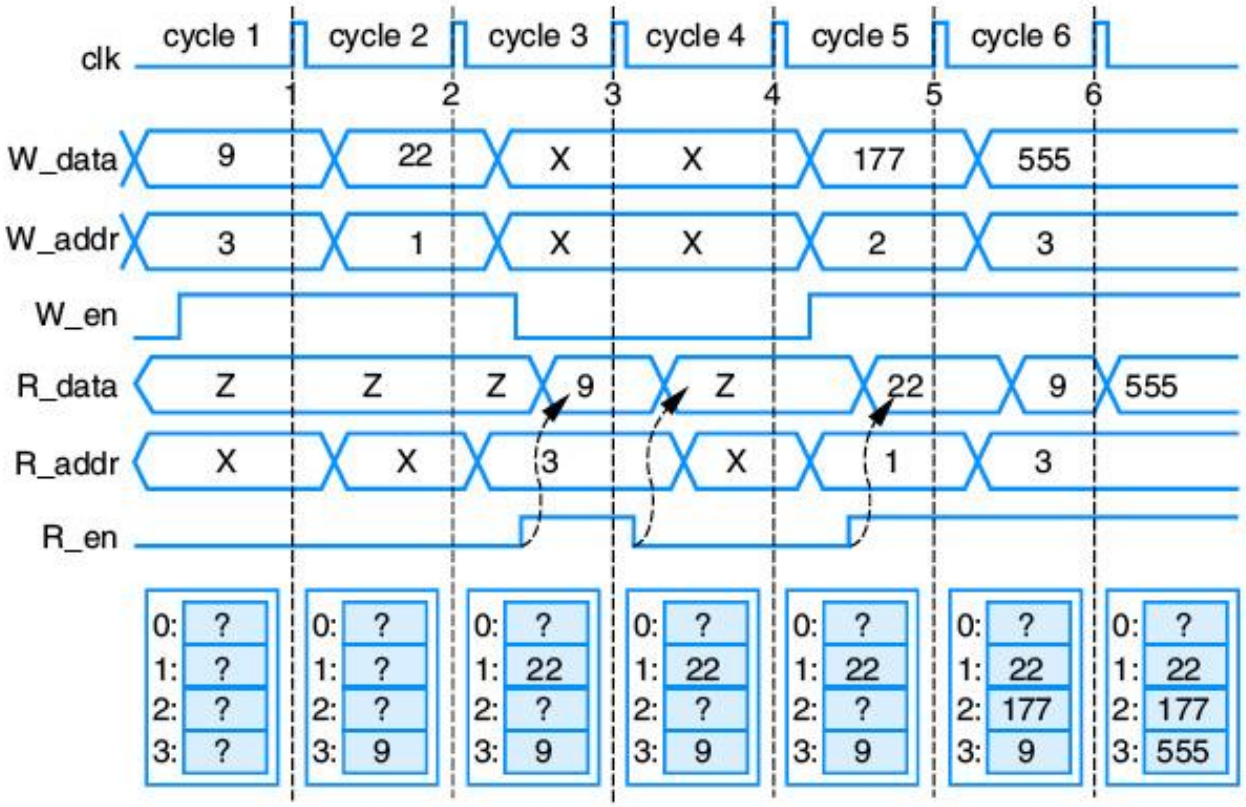


Figure 4.83 Writing and reading a register file.

Banco de Registradores MxN

- Tipos de Banco de Registradores
  - Banco de registradores de porta dupla:
    - 1 porta de leitura e 1 de escrita
  - Banco de registradores de porta simples:
    - Apenas 1 porta, usada tanto para ler como para escrever.
  - Banco de registradores de portas múltiplas:
    - 3 portas: uma porta de escrita e duas portas de leitura.
    - Especialmente útil em microprocessadores
- Instrução típica de microprocessador opera com dois registradores e armazena o resultado em um terceiro registrador, como na instrução
  - “RO ← R1 + R2”.

Banco de Registradores MxN

- Número típico de registradores:
  - De 4 a 1024,
- Larguras típicas:
  - de 8 a 64 bits por registrador
- Número de portas:
  - 1, 2, 3 portas e ate mais
- Indo além de três portas:
  - Desempenho do banco de registradores pode baixar e
  - Tamanho pode aumentar de forma significativa (dificuldade de roteamento)