

Chaveamento por Divisão no Tempo

- ⇒ No caso do chaveamento por divisão espacial em que são utilizados multi-estágios, o compartilhamento de *crosspoints* para mais do que uma conexão potencial provê significativa economia nos custos de implementação das chaves.
- ⇒ Nos casos estudados, os *crosspoints* de chaves multi-estágios por divisão no espaço são compartilhados de conexão à conexão, mas um *crosspoint* designado a uma conexão particular é dedicado àquela conexão durante toda a sua duração.
- ⇒ **Chaveamento por divisão no tempo** envolve o compartilhamento de *crosspoints* por períodos de tempo mais curtos, de tal forma que *crosspoints* individuais e seus *links* inter-estágios associados são continuamente re-designados a conexões existentes.
- ⇒ Este tipo de compartilhamento permite uma economia muito maior em *crosspoints*.
- ⇒ A economia é obtida pela multiplexação por divisão no tempo dos *crosspoints* e *links* inter-estágios, da mesma forma que os *links* de transmissão são compartilhados através de multiplexação por divisão no tempo para o compartilhamento de *links* entre centrais.
- ⇒ O chaveamento por divisão no tempo pode ser aplicado a sinais analógicos ou digitais.

Chaveamento Analógico por Divisão no Tempo

A Figura 3.41 ilustra uma estrutura simples de Chaveamento por Divisão no Tempo para sinais analógicos.

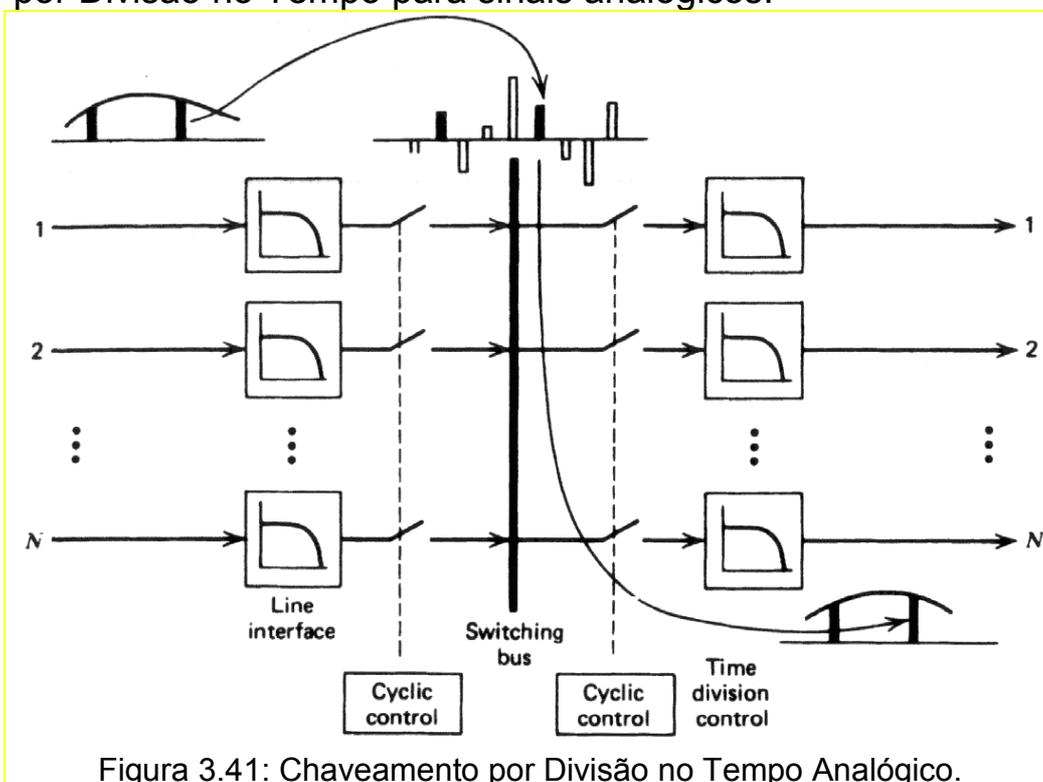


Figura 3.41: Chaveamento por Divisão no Tempo Analógico.

- ⇒ Um barramento único de chaveamento suporta um n° múltiplo de conexões através do intercalamento de amostras PAM recebidas pelas interfaces de linha. (Na técnica de modulação PAM - *Pulse Amplitude Modulation* - os sinais contínuos são representados por amostras tomadas no tempo, mas de amplitudes contínuas.)
- ⇒ O 1º registro de controle cíclico controla o acesso de entradas ao barramento, uma amostra de cada vez.
- ⇒ O 2º registro de controle opera em sincronismo com o 1º e seleciona a linha de saída apropriada para cada amostra de entrada.
- ⇒ Um conjunto completo de pulsos (um de cada linha de entrada ativa) é denominado *frame*.
- ⇒ A taxa de *frames* é igual à taxa de amostragem de cada linha. Para sinais de voz, a taxa de amostragem varia de 8 a 12 kHz.

Chaveamento Digital por Divisão no Tempo

A pesquisa original em chaveamento digital nos Laboratórios Bell foi reportada por Earle Vaughan em 1959.

Modelos de laboratório foram desenvolvidos para demonstrar o conceito de integrar sistemas de transmissão via multiplexação TDM digital com sistemas de chaveamento por divisão no tempo.

No entanto, os dispositivos eletrônicos não estavam suficientemente desenvolvidos, de tal forma que o desenvolvimento comercial do chaveamento digital não foi implementado.

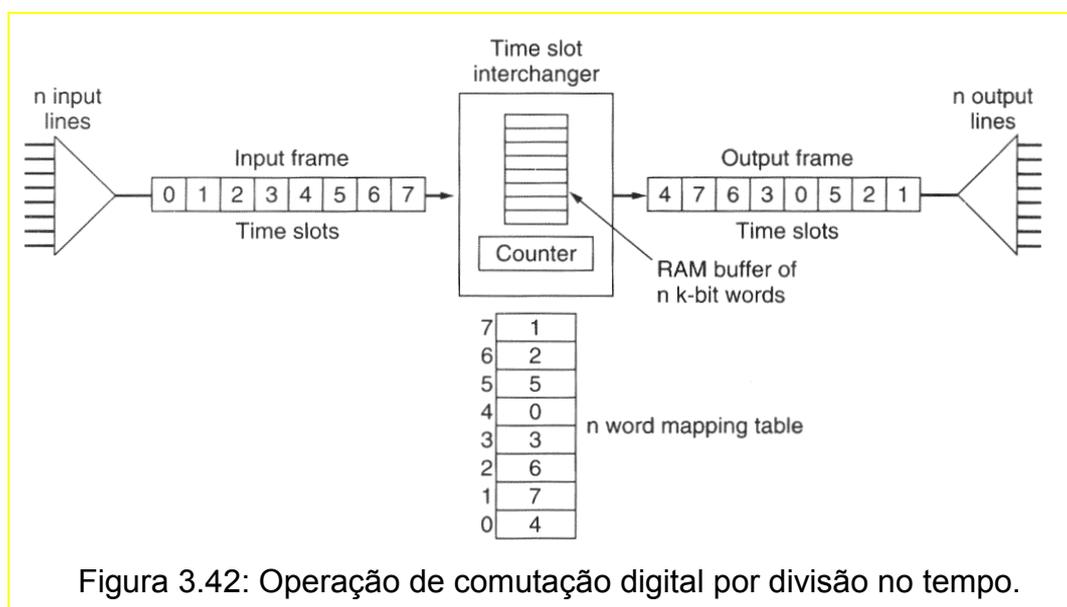
Por esta razão, o desenvolvimento da chave N° 1 ESS continuou seguindo a tecnologia eletromecânica.

Quase 10 anos depois, os Laboratórios Bell começaram o desenvolvimento de uma chave digital, denominada N° 4 ESS, para emprego em redes de centrais *toll* digitais.

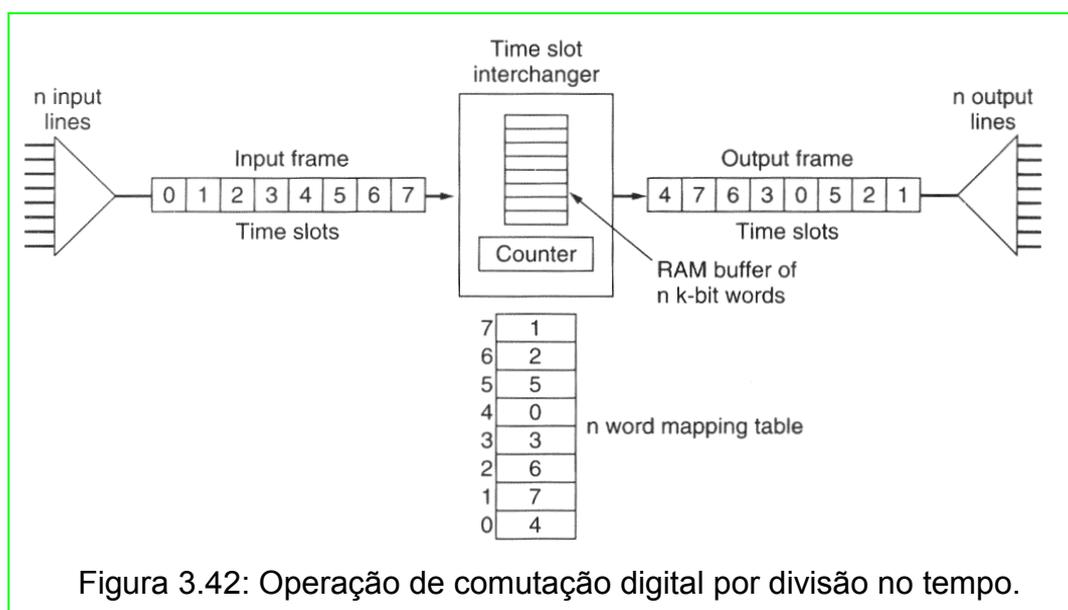
Em janeiro de 1976, a N° 4 ESS começou a operar, provendo várias novas capacidades para a rede de centrais *toll*:

- Foi a primeira chave *toll* a ser projetada para controle computadorizado.
- A capacidade da N° 4 ESS era três vezes a capacidade das chaves eletromecânicas existentes naquele tempo: as chaves *crossbar* N° 4 A.
- Assim, a principal característica da chave N° 4 ESS era que muitas áreas podiam consolidar o tráfego *toll* em uma chave ao invés de em várias.
- Por utilizar chaveamento por divisão no tempo, a chave digital N° 4 ESS permitiu a conexão direta com linhas T, eliminando os bancos de canais na interface entre troncos digitais e sistemas de chaveamento, resultando em economia significativa e melhora na qualidade da transmissão.

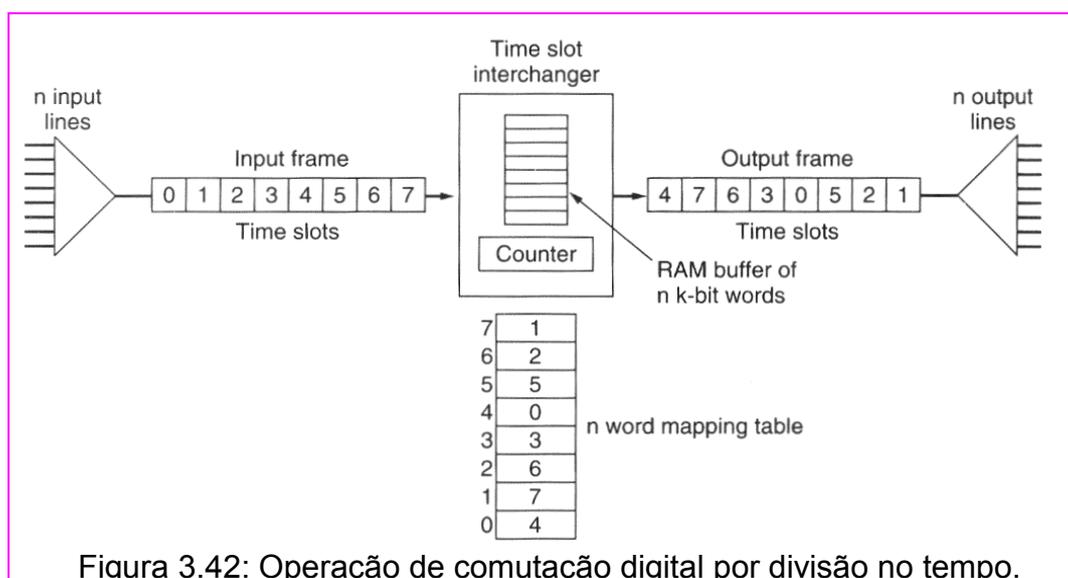
A essência funcional de uma operação de **chaveamento digital por divisão no tempo** é ilustrada na Figura 3.42.



- Todas as n entradas são *links* TDM.
- Estes *links* podem representar sistemas digitais com compartilhamento de linhas, troncos entre centrais em sistemas T ou as saídas de bancos de canais utilizados para interfacear linhas analógicas a centrais de comutação digital.
- Basicamente, a matriz de chaveamento deve transferir a informação que chega em um específico *time slot* (canal) de um *link* TDM para um específico *time slot* (canal) em um *link* TDM de saída.
- Na operação de chaveamento digital por divisão no tempo, as n linhas de entrada são varridas em seqüência para construir um *frame* de entrada com n slots.
- Cada *slot* é formado por k bits. (Para chaves T1, os *slots* são compostos por 8 bits, com 8000 *frames* processados por segundo).



- O coração da chave por divisão no tempo é um intercambiador de *time slots* (*time slot interchanger*).
- A função do intercambiador de *time slots* é receber *frames* de entrada e produzir *frames* de saída, nos quais os *slots* de tempo foram reordenados.
- Na figura 3.42, o *slot* de entrada 4 é conduzido à saída primeiro. Após, o *slot* de entrada 7 é conduzido à saída, e assim sucessivamente.
- Finalmente, o *frame* de saída é demultiplexado, com o *slot* de saída 0 (*slot* de entrada 4) saindo pela linha 0, o *slot* de saída 1 (*slot* de entrada 7) saindo pela linha 1, e assim sucessivamente.
- Em essência, a chave moveu 1 byte da linha de entrada 4 para a linha de saída 0, outro byte da linha de entrada 7 para a linha de saída 1, e assim sucessivamente.
- Visto de fora, o arranjo todo equivale a um chaveamento por circuito, embora não haja conexões físicas.



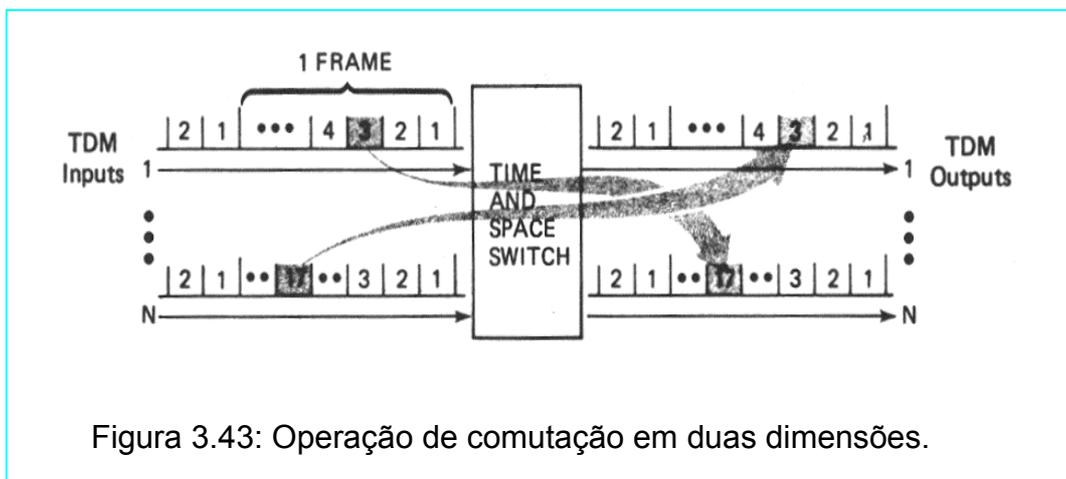
O intercambiador de *time slots* funciona da seguinte maneira:

- ⇒ Quando um *frame* de entrada está pronto para ser processado, cada *slot* (isto é, cada byte no *frame* de entrada) é escrito em um *buffer* RAM dentro do intercambiador.
- ⇒ Os *slots* são escritos em ordem.
- ⇒ Após todos os *slots* do *frame* de entrada terem sido armazenados no *buffer*, o *frame* de saída é construído através da leitura das palavras novamente, mas em uma ordem diferente.
- ⇒ Um contador é utilizado para contar de 0 até $n-1$.
- ⇒ No passo j , o conteúdo da palavra j de uma tabela de mapeamento é lido e usado para endereçar a tabela RAM.
- ⇒ Por exemplo, se a palavra 0 da tabela de mapeamento contém um 4, a palavra 4 do *buffer* RAM será lida primeiro, e o primeiro *slot* do *frame* de saída será o *slot* 4 do *frame* de entrada.
- ⇒ Desta forma, os conteúdos da tabela de mapeamento determinam qual permutação do *frame* de entrada será gerada à medida que o *frame* é entregue na saída, e conseqüentemente, qual linha de saída é conectada à qual linha de entrada.

- ⇒ As chaves por divisão no tempo usam tabelas que são lineares com respeito ao número de linhas de entrada, e não quadráticas, mas apresentam uma outra limitação.
- ⇒ É necessário armazenar n slots no *buffer* RAM e, então, lê-los novamente dentro de um período de *frame* equivalente a $125 \mu\text{s}$.
- ⇒ Se cada um destes acessos à memória leva $T \mu\text{s}$, o tempo necessário para processar um *frame* será $2nT \mu\text{s}$, de tal forma que teremos $2nT = 125$, ou $n = 125/2T$.
- ⇒ Para uma memória com um tempo de ciclo de 100 ns , será possível suportar no máximo 625 linhas
$$\left(n = \frac{125 \times 10^{-6}}{2 \times 100 \times 10^{-9}} = 625 \right).$$
- ⇒ Esta relação pode ser usada para determinar o ciclo de memória requerido para suportar um determinado número de linhas.
- ⇒ Como em uma chave *crossbar*, é possível derivar chaves de multi-estágios que dividam o trabalho em várias partes e então combinem os resultados em ordem para lidar com maiores quantidades de linhas.

Chaveamento em Duas Dimensões

- Chaves digitais maiores requerem operações de chaveamento tanto na dimensão espaço, quanto na dimensão tempo.
- Neste tipo de chave, uma dada conexão envolve dois *links* físicos diferentes e dois *time slots* diferentes.
- O processo de chaveamento requer:
 - translação espacial (chaveamento espacial) e
 - translação no tempo (chaveamento no tempo).
- Assim, a operação básica é referida como chaveamento em duas dimensões.
- As Figuras 3.43 e 3.44 apresentam estruturas simples de chaveamento temporal-espacial (chaveamento TS).



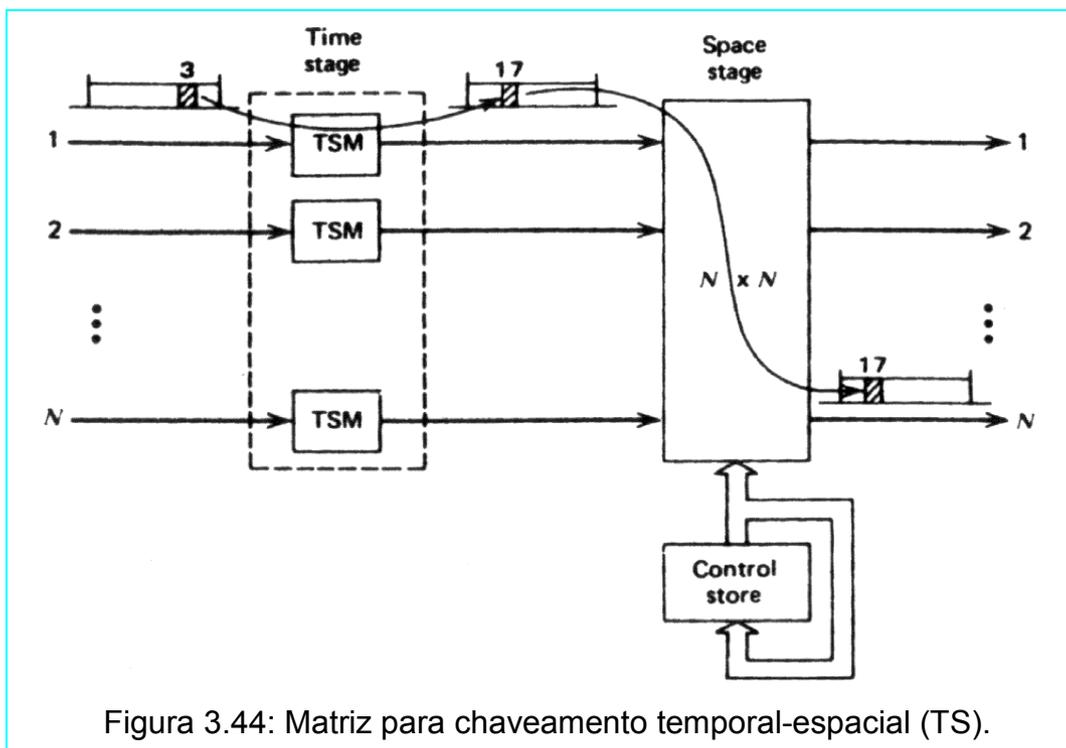
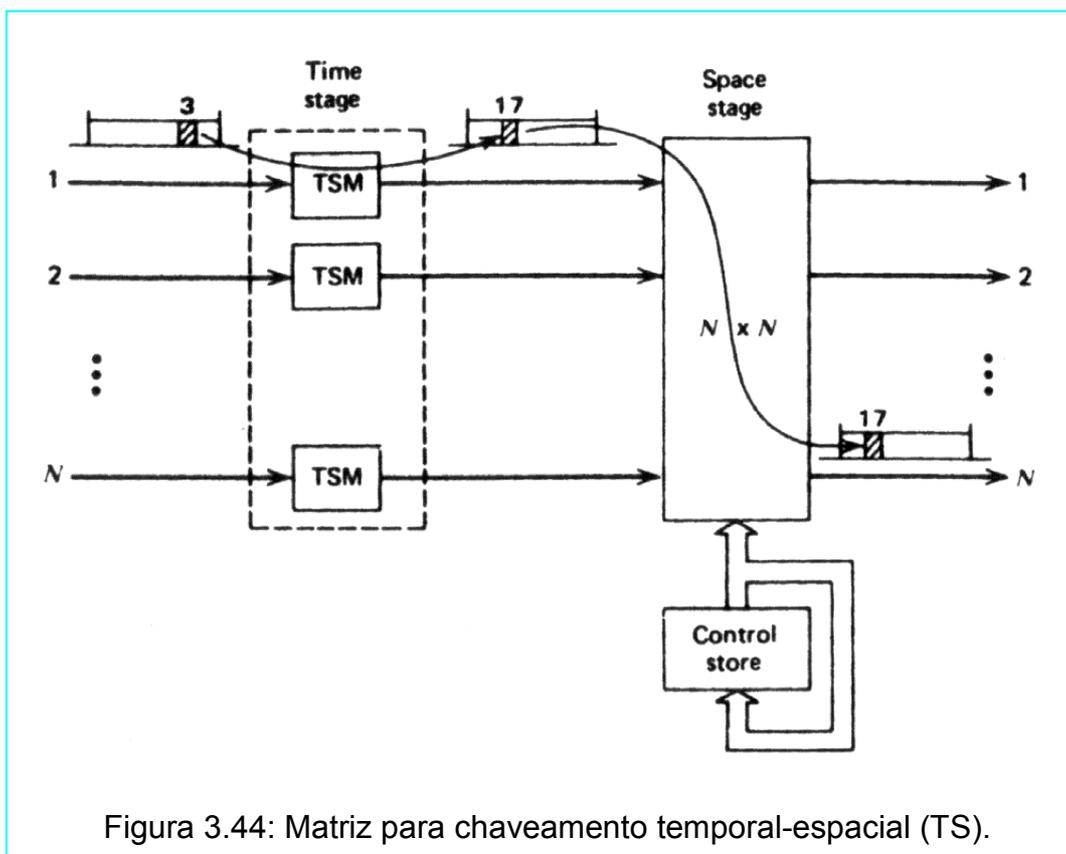


Figura 3.44: Matriz para chaveamento temporal-espacial (TS).

- * A chave para duas dimensões é constituída de dois estágios: um estágio temporal T seguido de um estágio espacial S.
- * A função básica do estágio temporal é atrasar a informação nos *time slots* de chegada até que ocorra o *time slot* desejado de saída.
- * Neste momento a informação atrasada é transferida através do estágio espacial para o *link* de saída apropriado.
- * O chaveamento espacial é obtido com circuitos seletores lógicos convencionais.
- * O chaveamento temporal é obtido através do armazenamento temporário da informação em uma memória digital ou circuito registrador.



- ⇒ A informação no *time slot* 3 de chegada do *link* é atrasada até que o *time slot* 17 de saída ocorra.
- ⇒ O caminho de retorno requer que a informação que chega no *time slot* 17 do *link* N seja atrasada para o *time slot* 3 do próximo *frame* de saída.
- ⇒ Observe que um estágio temporal pode ter que prover atrasos que irão de 1 *time slot* até 1 *frame* completo.
- ⇒ Associado com o estágio espacial está uma memória de controle que contém a informação necessária para especificar a configuração do estágio espacial para cada *time slot* individual de um *frame*.
- ⇒ Esta informação de controle é acessada ciclicamente da mesma forma que a informação de controle existente na chave analógica por divisão do tempo.
- ⇒ Por exemplo, para cada *time slot* de saída, a informação de controle é acessada para especificar qual *link* inter-estágio é conectado a qual *link* de saída.

Complexidade de Implementação de Chaves de Duas Dimensões

- No caso das chaves por divisão no espaço, o principal aspecto a considerar é a redução do nº de *crosspoints* e a decorrente influência na qualidade do serviço prestado (probabilidade nula ou não nula de ocorrer bloqueio de chamadas).
- No caso de chaves implementadas por divisão no tempo, o nº de *crosspoints* não é o parâmetro mais relevante a ser considerado.
- Em tais chaves, a complexidade de implementação passa a ser o parâmetro preponderante.
- Diferentes projetos podem ser implementados para uma chave de mesmo tamanho, com idênticas características.
- Além do nº de *crosspoints*, em uma chave digital, a memória total necessária deve ser incluída na estimativa do custo global.
- A memória requerida inclui um determinado nº de bits para as matrizes do estágio temporal e um determinado nº de bits requeridos para os registros de controle, tanto do estágio temporal quanto do estágio espacial.
- Em chaves grandes, o nº de *crosspoints* do estágio espacial pode ser reduzido às custas do aumento da quantidade de memória requerida.
- A complexidade de implementação é expressa como:

$$\text{Complexidade} = N_X + \frac{N_B}{100} \quad (3.43)$$

onde:

N_X = nº de *crosspoints* do estágio espacial

N_B = nº de bits de memória

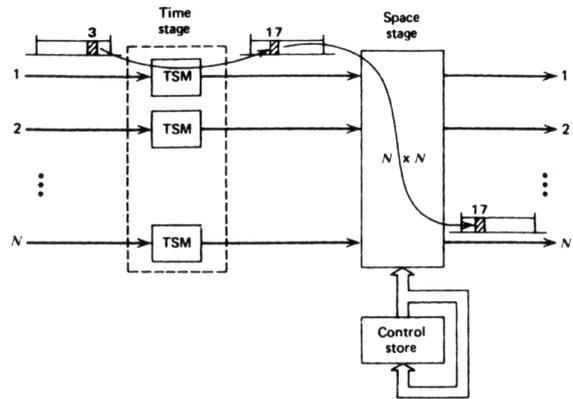
- Na expressão (3.43) o fator 100 representa uma relação de custo padrão (para diferentes projetos de chaves) de aproximadamente 1 *crosspoint* para 100 bits de memória.

Exemplo 3.1:

Determine a complexidade de implementação de uma chave TS de um estágio, conforme mostrada na Figura 3.44, onde o nº de linhas de entrada TDM é $N = 80$.

Assuma cada linha de entrada como contendo um sinal DS1 único (de 24 canais).

Figura 3.44: Matriz para chaveamento temporal-espacial (TS).



Solução:

O nº de *crosspoints* no estágio espacial é determinado por

$$N_X = 80^2 = 6400 \quad (3.44)$$

Para a determinação do nº de *crosspoints* é considerado N^2 porque são incluídas as auto-conexões e os pontos de cruzamentos fora da diagonal principal, o que é usual em *links* TDM.

O nº total de bits de memória para o registro de controle do estágio espacial é determinado conforme

$$N_{BX} = (\text{nº de links})(\text{nº de palavras de controle})(\text{nº de bits/palavra de controle})$$

$$N_{BX} = (80)(24)(7) = 13400 \quad (3.45)$$

O nº de bits de memória no estágio temporal é determinado como a soma dos bits do intercambiador de *time slots* e dos bits de registro de controle, assim

$$N_{BT} = (\text{nº de links})(\text{nº de canais})(\text{nº de bits/canal}) +$$

$$+ (\text{nº de links})(\text{nº de palavras de controle})(\text{nº de bits/palavra de controle})$$

$$N_{BT} = (80)(24)(8) + (80)(24)(5) = 24960 \quad (3.46)$$

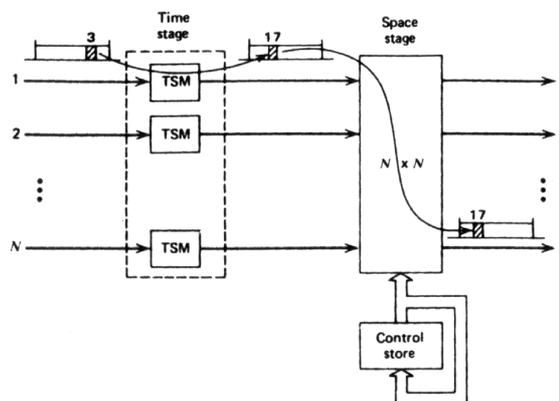
Então, a complexidade da implementação é determinada por (3.47)

$$\text{Complexidade} = N_X + \frac{N_{BX} + N_{BT}}{100} = 6784 \text{ crosspoints equivalentes}$$

* Note que a determinação da complexidade de implementação é dominada pelo nº de *crosspoints* no estágio espacial.

- * Uma complexidade significativamente baixa (e geralmente um custo inferior) podem ser obtidos se grupos de *links* de entrada são combinados em sinais multiplexados de níveis superiores, antes do chaveamento.
- * Uma limitação significativa da estrutura TS mostrada na Figura 3.44 ocorre quando uma conexão precisa ser feita para um canal específico de uma saída (ao invés de a qualquer canal de uma saída).
- * Um conexão a um canal específico é bloqueada sempre que o desejado *slot* de tempo do *link* TDM já está em uso.

- * Por exemplo, se o *time slot* 17 do 1º *link* TDM está em uso para alguma conexão que parta da 1ª entrada para ser conectada a algum *link* diferente do *link* N, a conexão do canal 3 da entrada 1 para o canal 17 da entrada N não pode ser estabelecida.



- * Assim, a estrutura TS é útil apenas se as saídas representam grupos de troncos, o que implica que qualquer canal de saída é adequado.
- * Uma conexão *full-duplex* requer que a conexão reversa seja estabelecida, o que adiciona ainda mais restrições a quais canais de saída podem ser usados.
- * Aplicações que demandam conexões a canais específicos requerem estágios adicionais para que seja atingido um desempenho adequado com relação a probabilidades de bloqueio.

Chaves Multi-Estágios em Duas Dimensões

- ⇒ Uma forma de reduzir o custo de uma chave por divisão no tempo é multiplexar tantos canais quanto possíveis e desempenhar tanto chaveamento nos estágios temporais quanto possível.
- ⇒ Chaveamento no estágio temporal é geralmente mais barato do que chaveamento no estágio espacial - a relação de custo entre os *crosspoints* digitais é de aproximadamente 1 *crosspoint* para 100 bits de memória.
- ⇒ Naturalmente há limites práticos ao n^o de canais que podem ser multiplexados em um *link* TDM comum, para o estágio de chaveamento temporal.
- ⇒ Quando estes limites práticos são atingidos, uma maior redução na complexidade de implementação (e conseqüentemente nos custos) pode ser obtida apenas através do uso de múltiplos estágios.
- ⇒ Uma implementação geralmente mais efetiva envolve:
 - 1. Chaves Espaço-Tempo-Espaço (STS)**, onde dois estágios espaciais são separados por um estágio temporal e
 - 2. Chaves Tempo-Espaço-Tempo (TST)**, onde dois estágios temporais são separados por um estágio espacial.

Chaves STS

Um diagrama funcional de uma chave STS é mostrado na Figura 3.45.

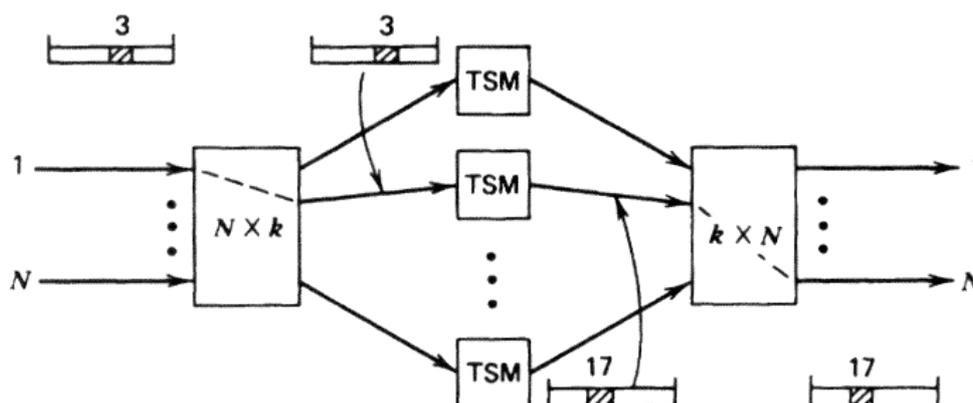


Figura 3.45: Estrutura de chaveamento Espaço-Tempo-Espaço (STS).

- Cada chave espacial é constituída de um único estágio, sem oferecer bloqueio.
- Para chaves muito grandes, pode ser desejável implementar as chaves espaciais com múltiplos estágios.
- O estabelecimento de um caminho por uma chave STS requer determinar uma matriz de chaveamento temporal com um acesso para escrever disponível no *time slot* que chega e um acesso para leitura disponível no *time slot* desejado de saída.
- Quando cada estágio individual (S, T, S) não permite bloqueio, a operação é funcionalmente equivalente à operação de uma chave espacial de três estágios.

- Assim, o grafo de probabilidades mostrado na Figura 3.46 de uma chave STS é idêntico ao grafo de probabilidade para uma chave de 3 estágios mostrada na Figura 3.38.

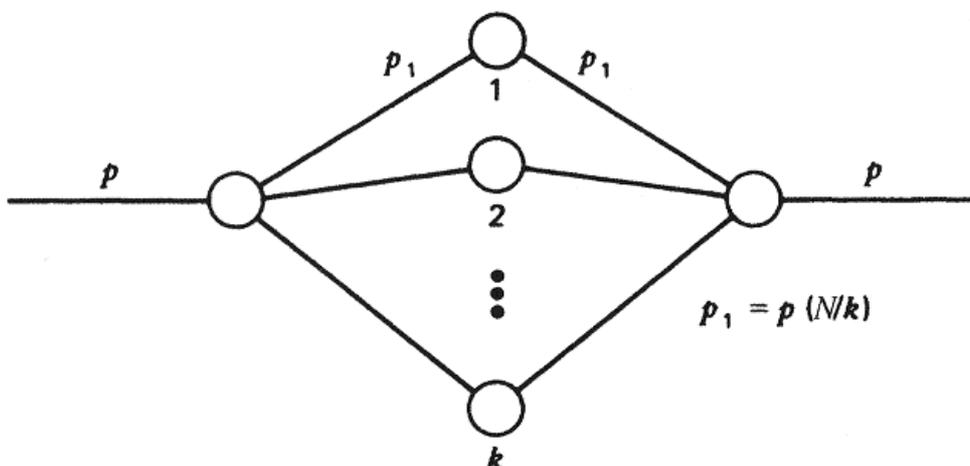


Figura 3.46: Grafo de probabilidade de uma chave STS com estágios que não bloqueiam.

- De forma correspondente, a probabilidade de bloqueio de uma chave STS é dada por

$$B = (1 - q'^2)^k \quad (3.48)$$

onde:

k é o nº de matrizes de chaveamento temporal do estágio central e $q' = 1 - p' = 1 - \frac{p}{\beta}$, para $\beta = \frac{k}{N}$.

- Assumindo as chaves espaciais como sendo matrizes de um único estágio e que cada *link* TDM tenha c canais, pode-se determinar a complexidade da implementação de uma chave STS como

Complexidade = nº de *crosspoints* do estágio espacial +
 (nº de bits de controle do estágio espacial +
 nº de bits de memória do estágio temporal +
 nº de bits de controle do estágio temporal)/100

$$\text{Complexidade} = 2kN + \frac{2kc \log_2 N + kc(8) + kc \log_2 c}{100} \quad (3.49)$$

Exemplo 3.2:

Determine a complexidade de implementação de uma chave STS de 2048 canais para 16 *links* TDM com 128 canais em cada *link*. A probabilidade de bloqueio máxima desejada é 0.002 para uma ocupação de canal de 0.1.

Solução:

O nº mínimo de chaves temporais de estágio central para prover o grau de serviço desejado pode se determinado a partir da Equação 3.48 como sendo $k = 7$. Em (3.48),

$$B = (1 - q'^2)^k = \left(1 - \left(1 - \frac{pN}{k} \right)^2 \right)^k, \text{ assim,}$$

$$0.002 = \left(1 - \left(1 - \frac{(0.1)(16)}{k} \right)^2 \right)^k, \text{ de onde } k = 7.191 \cong 7.$$

Usando o valor determinado para k , o número de *crosspoints* é determinado como sendo $2kN = (2)(7)(16) = 224$.

O número de bits de memória pode ser determinado como

$$\begin{aligned} & 2kc \log_2 N + kc(8) + kc \log_2 c = \\ & = (2)(7)(128) \log_2 (16) + (7)(128)(8) + (7)(128) \log_2 (128) = 20608 \end{aligned}$$

Assim, a complexidade composta da implementação é igual a

$$\begin{aligned} \text{Complexidade} &= 2kN + \frac{2kc \log_2 N + kc(8) + kc \log_2 c}{100} = \\ &= 224 + \frac{20608}{100} = 430 \text{ crosspoints} \end{aligned}$$

O valor obtido para a complexidade da implementação (430 *crosspoints*) deve ser comparado ao número de *crosspoints* obtidos para uma chave de 3 estágios, de tamanho equivalente, listada na Tabela 3.6.

Tabela 3.6					
Tamanho da chave, N	n	k	β	Nº de <i>crosspoints</i>	Nº de <i>crosspoints</i> para chaves sem bloqueio
...
2048	32	10	0.313	81920	516096 ($k=63$)
...
A chave assume que as entradas estão ocupadas apenas 10%.					

Observe dos dados transcritos da Tabela 3.6 que o projeto da chave espacial requer 81920 *crosspoints* enquanto que o projeto STS requer apenas 430 *crosspoints* equivalentes.

Chaves TST

A segunda forma de chave espaço-tempo de múltiplos estágios, a chave TST, é mostrada na Figura 3.47.

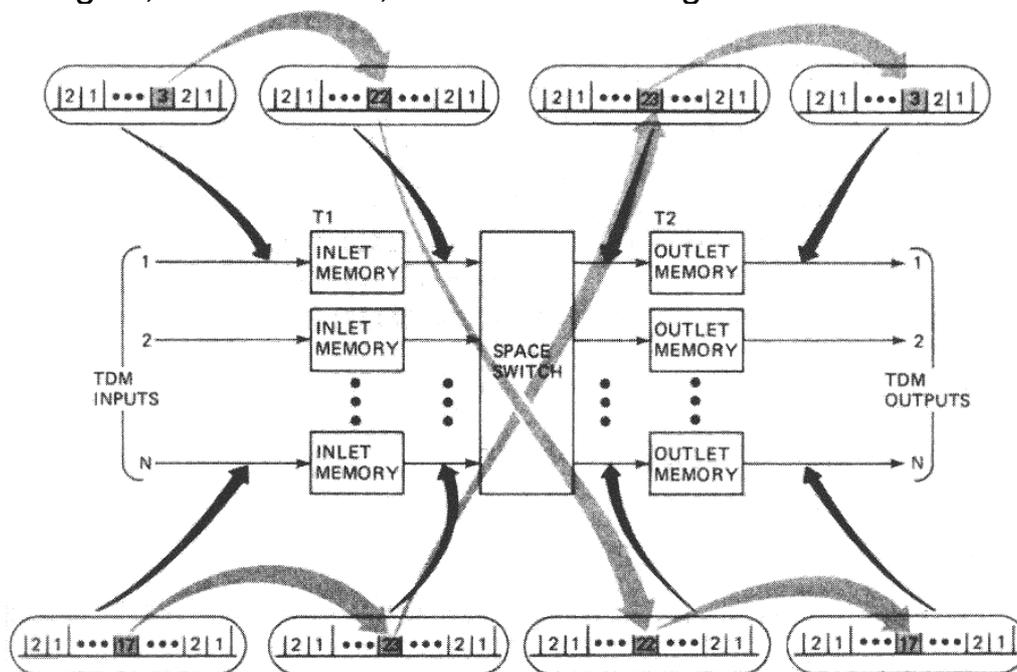


Figura 3.47: Estrutura de chaveamento Tempo-Espaço-Tempo (TST).

- A informação vinda de um canal TDM por um *link* de chegada é atrasada no estágio temporal de entrada até que um caminho apropriado através do estágio espacial esteja disponível.
- Neste ponto a informação é transferida através do estágio espacial ao estágio temporal de saída apropriado onde é retida até que ocorra o *time slot* de saída desejado.
- Assumindo que todos os canais que chegam podem ser conectados a todos os canais de saída, qualquer *time slot* do estágio espacial pode ser usado para estabelecer uma conexão.
- De forma funcional o estágio espacial é replicado uma vez para cada *time slot* interno.

- Este conceito é reforçado pelo grafo de probabilidade TST mostrado na Figura 3.48.

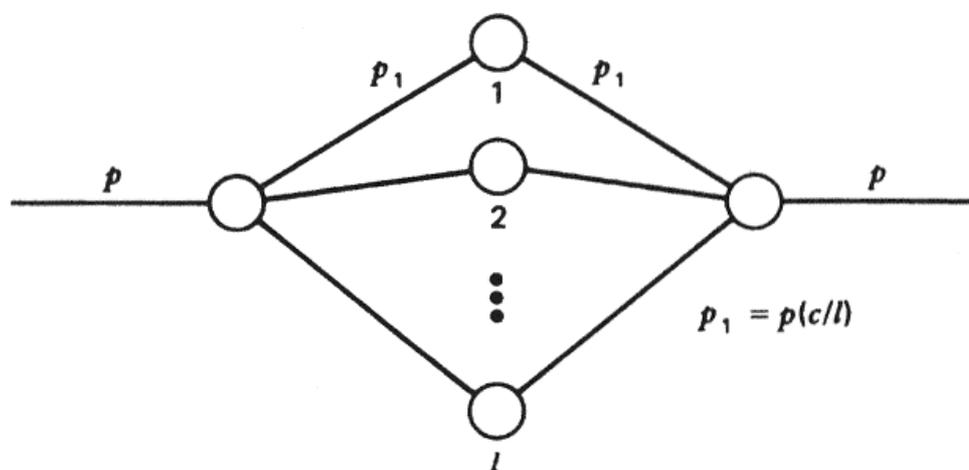


Figura 3.48: Grafo de probabilidade de uma chave TST com estágios que não bloqueiam.

- Uma característica importante de uma chave TST é que o estágio espacial opera de uma forma dividida no tempo, independentemente dos *links* externos TDM.
- Na verdade, o n° l de *time slots* do estágio espacial não tem que coincidir com o n° c de *time slots* externos TDM.
- Se o estágio espacial não permite bloqueios, o bloqueio em uma chave TST ocorre apenas se não há *time slot* de estágio espacial interno durante o qual o *link* entre o estágio temporal de entrada e o estágio temporal de saída estão ambos desocupados.
- Obviamente, a probabilidade de bloqueio é minimizada se o n° de *time slots* l do estágio espacial é feito grande.
- Na verdade, como uma analogia direta das chaves espaciais de 3 estágios, a chave TST é estritamente não bloqueadora se $l = 2c - 1$.

- A expressão geral da probabilidade de bloqueio para uma chave TST com estágios individuais que não bloqueiam (T, S, T) é

$$B = (1 - q_1^2)^Y \quad (3.50)$$

onde:

$$q_1 = 1 - p_1 = 1 - \frac{p}{\alpha}$$

l = n° de *time slots* do estágio espacial

α = expansão temporal ($\alpha = l/c$)

A complexidade da implementação de uma chave TST pode ser derivada como

$$\text{Complexidade} = N^2 + \frac{Nl \log_2 N + 2Nc(8) + 2Nl \log_2 c}{100} \quad (3.51)$$

Exemplo 3.3:

Determine a complexidade de implementação de uma chave TST de 2048 canais com 16 *links* TDM com 128 canais por *link*.

A probabilidade de bloqueio máxima desejada é 0.002 para uma ocupação de canal de 0.1.

Solução:

Através da Equação (3.50), pode-se determinar o n° de *time slots* internos requerido para o grau de serviço desejado como sendo $l = 25$. Em (3.50),

$$B = (1 - q_1^2)^l = \left(1 - \left(1 - \frac{p}{\alpha}\right)^2\right)^l = \left(1 - \left(1 - \frac{cp}{l}\right)^2\right)^l, \text{ assim,}$$

$$0.002 = \left(1 - \left(1 - \frac{(128)(0.1)}{l}\right)^2\right)^l, \text{ de onde } l = 24.313 \cong 25.$$

Para $l = 25$, uma concentração temporal $1/\alpha = 5.12$ é possível devido à carga leve ($p = 0.1$) sobre os canais de entrada ($1/\alpha = c/l = 128/25 = 5.12$).

Usando o valor encontrado para l , o número de *crosspoints* poder ser determinado através da Equação (3.51). Assim,

$$\begin{aligned} \text{Complexidade} &= N^2 + \frac{Nl \log_2 N + 2Nc(8) + 2Nl \log_2 c}{100} = \\ &= (16)^2 + \frac{(16)(25) \log_2 (16) + 2(16)(128)(8) + 2(16)(25) \log_2 (128)}{100} = 656 \end{aligned}$$

Observe, nos Exemplos 3.2 e 3.3, que o valor de complexidade encontrado para a chave TST (656 *crosspoints*) é superior ao valor encontrado para a chave STS (430 *crosspoints*).

- ⇒ A chave TST opera com concentração temporal, enquanto que a chave STS opera com concentração espacial.
- ⇒ À medida que a utilização dos *links* de entrada aumenta, menos concentração é aceitável.
- ⇒ Se a carga do canal de entrada é alta, serão requeridas **expansão no tempo na chave TST** e **expansão no espaço na chave STS** para manter baixas probabilidades de bloqueio.
- ⇒ **Como a expansão no tempo pode ser obtida a um custo menor do que a expansão espacial, uma chave TST se torna mais eficiente com relação a custo do que uma chave STS, para canais com alta utilização.**

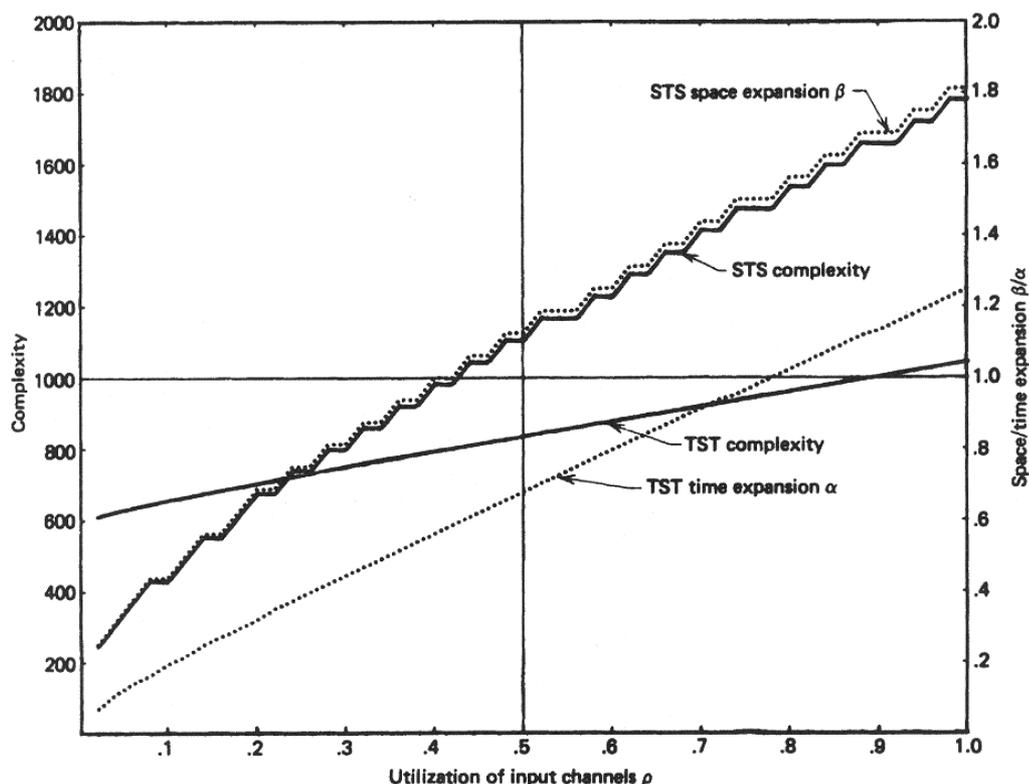


Figura 3.49: Comparação da complexidade de estruturas de chaveamento STS e TST para $B=0.002$, como função da utilização dos canais de entrada.

- ⇒ Chaves TST têm vantagem de implementação sobre chaves STS quando tráfego elevado está presente. Para chaves pequenas, a complexidade de implementação favorece as arquiteturas STS.
- ⇒ Um exemplo deste fato é a chave N° 4 ESS, uma estrutura TST (1976) para atender a 100000 canais de voz.

Chaves TSSST

- ⇒ Quando o estágio espacial de uma chave TST é grande o suficiente para justificar complexidade de controle adicional, estágios espaciais múltiplos podem ser usados para reduzir a quantidade total de *crosspoints*.
- ⇒ A Figura 3.50 descreve uma arquitetura TST com uma chave espacial de 3 estágios.
- ⇒ Na chave mostrada na Figura, como os 3 estágios intermediários são todos estágios espaciais, esta estrutura é chamada chave TSSST.
- ⇒ Um exemplo de chave TSSST é a chave EWSD da Siemens.

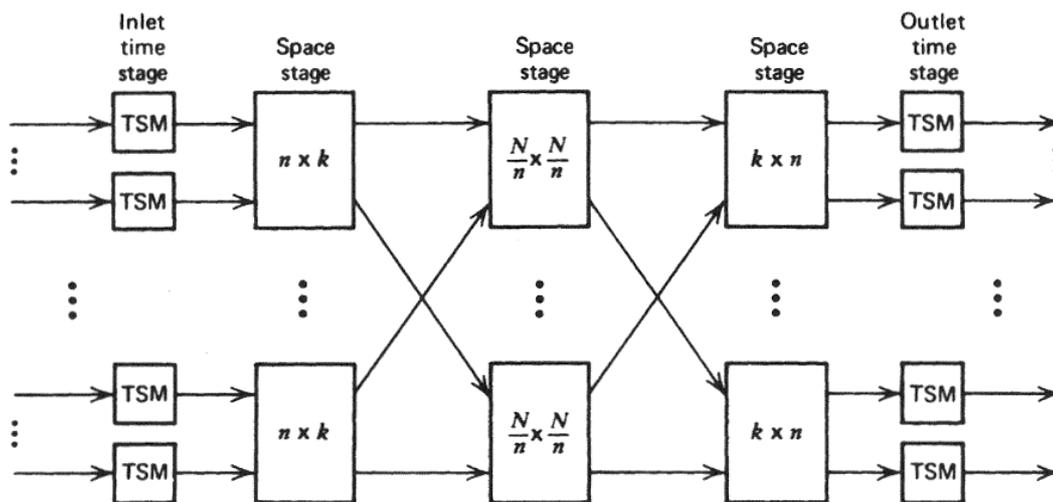


Figura 3.50: Estrutura de chaveamento TSSST.

A complexidade da implementação de uma chave TSSST pode ser determinada pela Equação (3.52).

$$\text{Complexidade} = N_X + \frac{N_{\text{BX}} + N_{\text{BT}} + N_{\text{BTC}}}{100} \quad (3.52)$$

onde:

$$N_X = \text{n}^\circ \text{ de crosspoints} = 2Nk + k(N/n)^2$$

$$N_{\text{BX}} = \text{n}^\circ \text{ de bits de registro de controle do estágio espacial} = \\ = 2k(N/n)l \log_2(n) + k(N/n)l \log_2(N/n)$$

$$N_{\text{BT}} = \text{n}^\circ \text{ de bits nos estágios temporais} = 2Nc(8)$$

$$N_{\text{BTC}} = \text{n}^\circ \text{ de bits de registro de controle do estágio temporal} \\ = 2Nl \log_2(c)$$

- * O grafo de probabilidades de uma chave TSSST é mostrado na Figura 3.50.
- * Note que o diagrama é funcionalmente idêntico ao grafo de probabilidades de uma chave de 5 estágios mostrada na Figura 3.40.

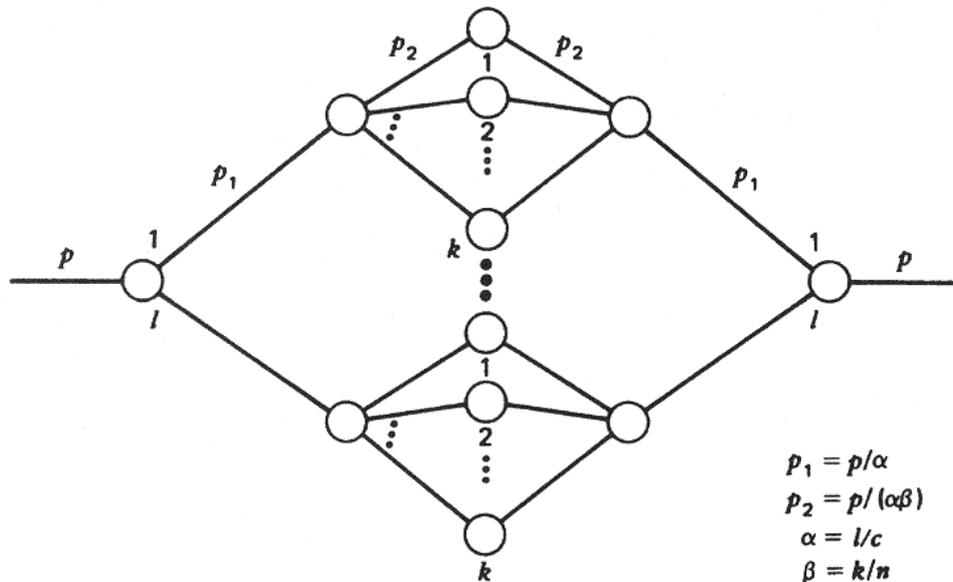


Figura 3.50: Grafo de probabilidade de uma chave TSSST.

- * Usando o grafo de probabilidades de uma chave TSSST, mostrado na Figura 3.50, a probabilidade de bloqueio pode ser determinada por

$$B = \left\{ 1 - (q_1)^2 \left[1 - (1 - q_2^2)^k \right] \right\} \quad (3.53)$$

onde:

$$q_1 = 1 - p_1 = 1 - \frac{p}{\alpha} \quad \text{e} \quad q_2 = 1 - p_2 = 1 - \frac{p}{\alpha\beta}$$

Exemplo 3.4:

Determine a complexidade de implementação de uma chave TSSST de 131072 canais, projetada para prover máxima probabilidade de bloqueio de 0.002, para uma ocupação de canais de entrada de 0.7.

Assuma que a chave atenda 1024 *links* de entrada TDM com 128 canais (*time slots*) por *link*.

Assuma, ainda, que é usada expansão unitária no tempo ($l = c \therefore \alpha = 1$) nos estágios espaciais.

Solução:

A chave espacial pode ser projetada de várias formas, dependendo de quantos *links* são atribuídos a cada matriz no primeiro (e no terceiro) estágios espaciais.

Anteriormente mostramos que derivando a Equação (3.33) (usada para a determinação do número de crosspoints N_X de chaves espaciais de 3 estágios que não permitam bloqueios) com respeito a n e igualando a derivada a zero é obtido o valor ótimo para n , valor que conduzirá ao n° mínimo de *crosspoints*.

O valor teórico ótimo para n é expresso por $n = \sqrt{N/2}$.

Desta forma, $n = \sqrt{1024/2} = 22.63$, conduzindo à escolha de $n = 32$, por ser o n° binário imediatamente superior a 22.63.

Tendo sido determinado n , a Equação (3.53) (para a probabilidade de bloqueio B) pode ser resolvida para a única incógnita restante, que é o n° de estágios centrais k .

Assim, é determinado $k = 27$, para uma probabilidade de bloqueio de 0.0015 ($0.0015 < (B_{\text{máxima}} = 0.002)$).

A complexidade pode, então, ser determinada a partir da Equação (3.52). Assim,

$$\text{Complexidade} = N_x + \frac{N_{\text{BX}} + N_{\text{BT}} + N_{\text{BTC}}}{100} \quad (3.52)$$

onde:

$$\begin{aligned} N_x &= \text{n}^\circ \text{ de } \textit{crosspoints} = \\ &= 2Nk + k(N/n)^2 = (2)(1024)(27) + 27(1024/32)^2 = 82944 \end{aligned}$$

$$\begin{aligned} N_{\text{BX}} &= \text{n}^\circ \text{ de bits de registro de controle do estágio espacial} = \\ &= 2k(N/n)l \log_2(n) + k(N/n)l \log_2(N/n) = \\ &= (2)(27)(1024/32)(128) \log_2(32) + \\ &\quad + (27)(1024/32)(128) \log_2(1024/32) = 1658880 \end{aligned}$$

$$N_{\text{BT}} = \text{n}^\circ \text{ de bits nos estágios temporais} = 2Nc(8) = (2)(1024)(8)$$

$$\begin{aligned} N_{\text{BTC}} &= \text{n}^\circ \text{ de bits de registro de controle do estágio temporal} \\ &= 2Nl \log_2(c) = (2)(1024)(128)(\log_2 128) = 1835008 \end{aligned}$$

Resultando em:

$$\text{Complexidade} = 82944 + \frac{1658880 + 2097152 + 1835008}{100} = 138854$$

crosspoints equivalentes.

Chaves Toll N° 4 ESS

A Figura 3.51 apresenta a estrutura básica de uma chave N° 4 ESS, que é do tipo TSSSST (isto é, uma matriz tempo-espaço-tempo com 4 estágios espaciais).

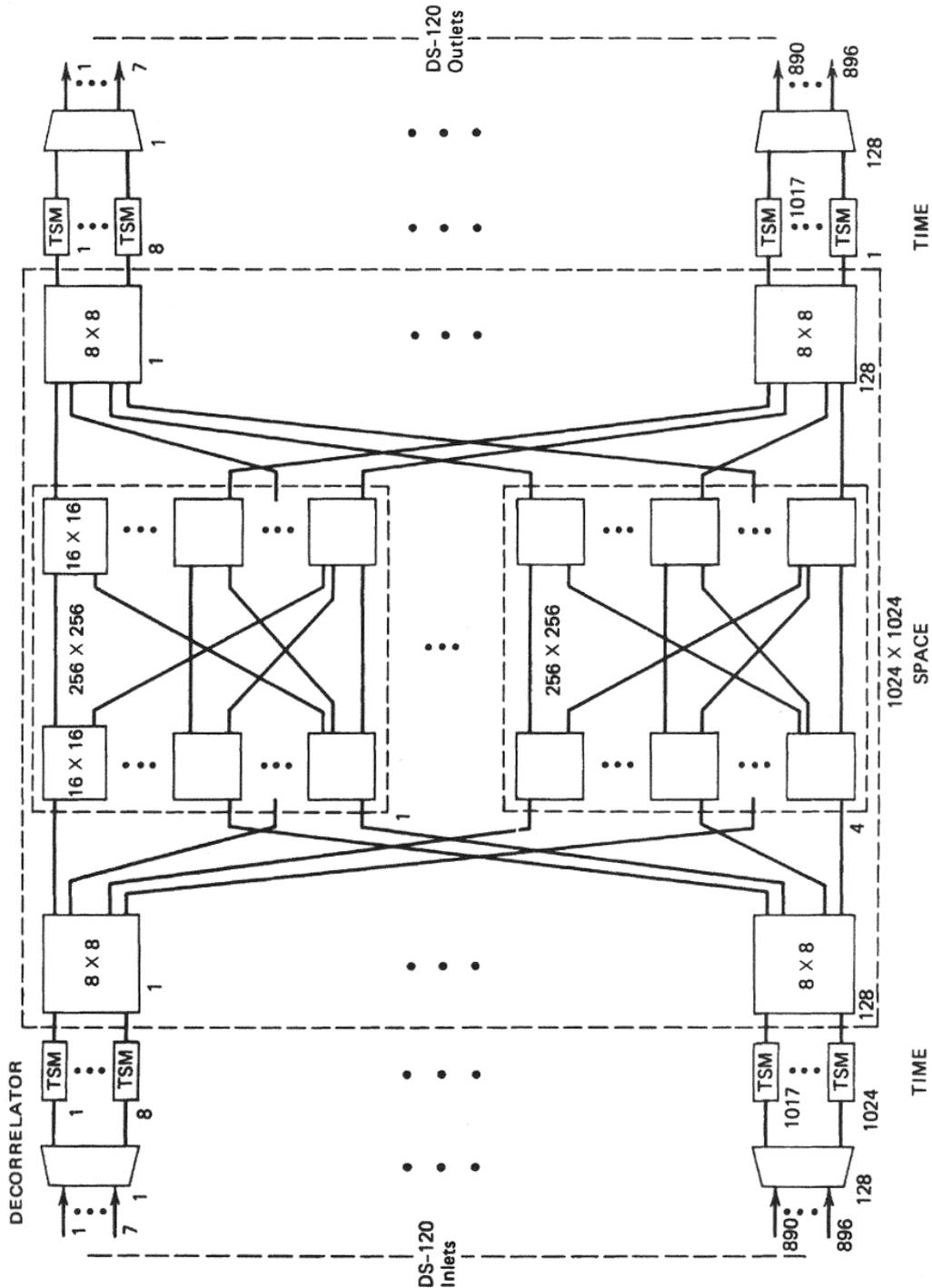


Figura 3.51: Arquitetura de uma matriz N° 4 ESS.

- ⇒ As entradas para a matriz são *links* TDM de 120 canais, formados pela multiplexação de 5 sinais DS1.
- ⇒ Antes destas entradas serem interfaceadas à matriz, elas são passadas através de um decorrelator para distribuir os canais de grupos troncos TDM através dos múltiplos *links* TDM da matriz.
- ⇒ A decorrelação precisa ser usada porque a chave N° 4 ESS é uma chave *toll* na qual os *links* TDM que chegam representam grupos tronco provenientes de outros dispositivos de comutação.
- ⇒ Em contraste a *links* TDM de uma chave de central local que são formados pela multiplexação de canais de assinantes independentes, os canais de um grupo tronco não são independentemente ocupados.
- ⇒ Na verdade, o ato de concentrar fontes múltiplas e independentes em um grupo tronco leva a altos níveis de correlação entre a atividade dos canais individuais.
- ⇒ Se os canais de um grupo tronco não forem decorrelacionados, poderão ocorrer probabilidades de bloqueio muito mais elevadas, porque todos os canais poderão estar disputando os mesmos caminhos através da matriz.
- ⇒ A decorrelação embaralha os grupos tronco de tal forma que caminhos alternados disponíveis para qualquer conexão particular terão maior probabilidade de serem estatisticamente independentes.
- ⇒ Observe na Figura que, além do embaralhamento dos canais, o decorrelator provê expansão espacial (de 7 para 8) e expansão temporal (de 120 para 128).
- ⇒ O tamanho máximo de uma chave N° 4ESS utiliza 128 decorreladores com 7 canais de 120 entradas cada. Assim, a capacidade máxima de canal é $(128)(7)(120) = 107520$ canais.
- ⇒ O estágio espacial é constituído por uma matriz de 1024×1024 com quatro possíveis caminhos para cada um dos 128 *time slots* dos estágios espaciais.

⇒ O grafo de probabilidades da chave N° 4ESS é mostrado na Figura 3.52.

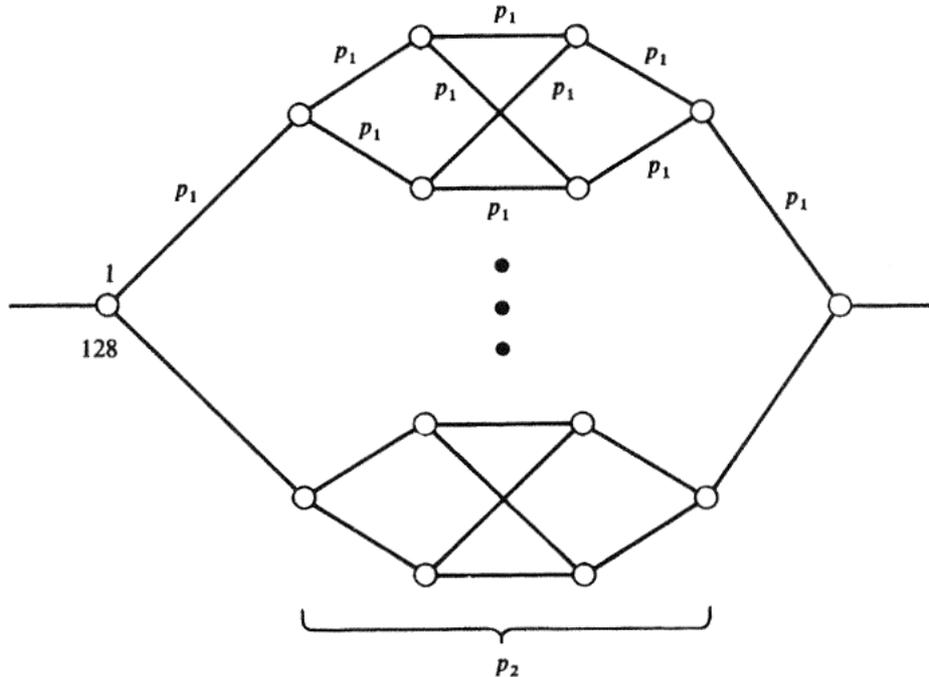


Figura 3.52: Grafo de probabilidades da chave N° 4ESS.

A partir do grafo mostrado na Figura 3.52 pode ser derivada a expressão para a probabilidade de bloqueio, conforme Equação (3.54).

$$B = [1 - (1 - p_1)(1 - p_2)(1 - p_1)]^{128} \quad (3.54)$$

onde:

p = utilização do canal externo,

$p_1 = (7/8)(120/128)p$ e

p_2 = probabilidade de bloqueio de uma chave espacial de 4 estágios, dada por:

$$p_2 = 2p_1^2q_1^6 + 16p_1^3q_1^5 + 50p_1^4q_1^4 + 52p_1^5q_1^3 + 28p_1^6q_1^2 + 8p_1^7q_1 + p_1^8 \quad (3.55)$$

⇒ Baixas probabilidades de bloqueio podem ser obtidas com a utilização de chaves N° 4ESS mesmo quando as entradas individuais são altamente utilizadas, devido ao elevado n° de caminhos alternativos existentes.