Chapter4 Pipelining Technology

Three methods for Improving the processing speed and system efficiency

- **Time-interleaving**
- **Resource-replication**
- **Resource-sharing**

Overview of Pipelining Technology

1. Sequence

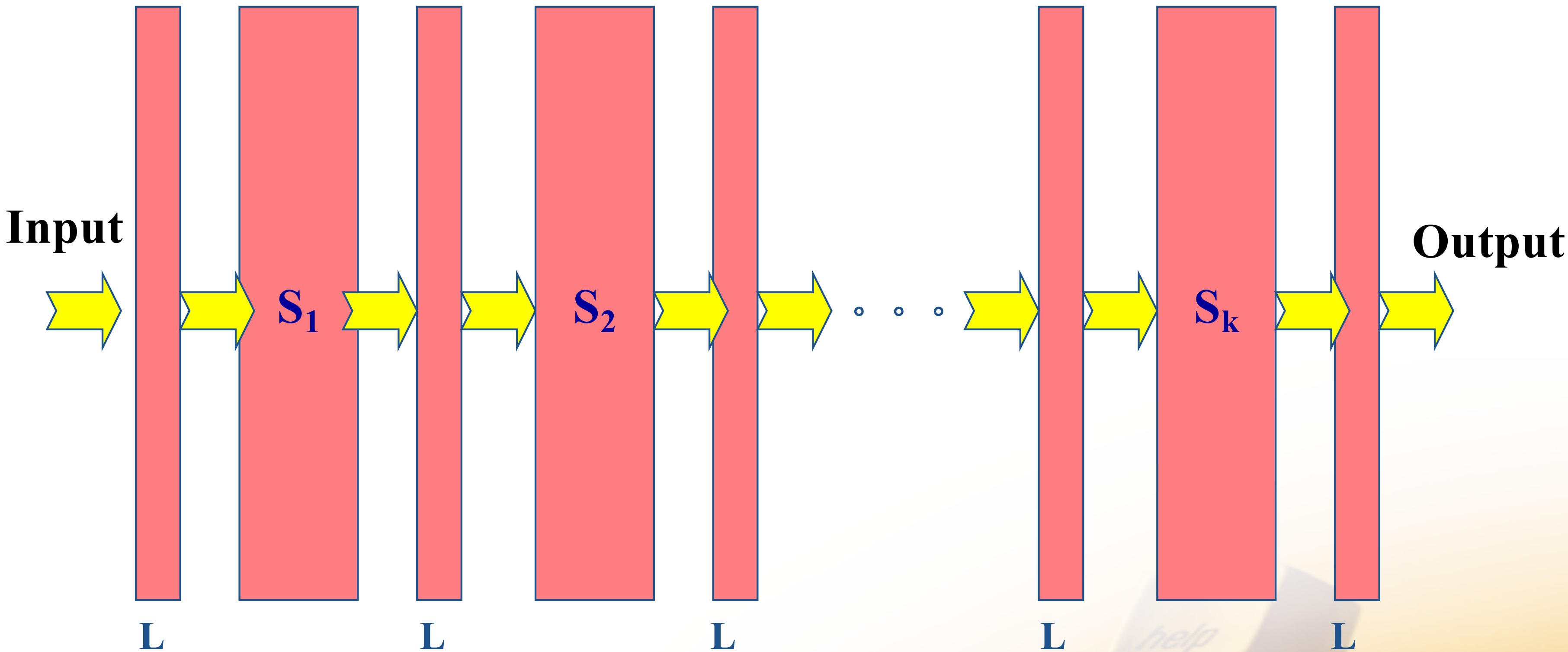
IF_k	ID_k	EX_k	IF_{k+1}	ID_{k+1}	EX_{k+1}	…

$$T = \sum_{i=1}^{n} \left(t_{\mathrm{IF}\ i} + t_{\mathrm{ID}\ i} + t_{\mathrm{EX}\ i} \right)$$

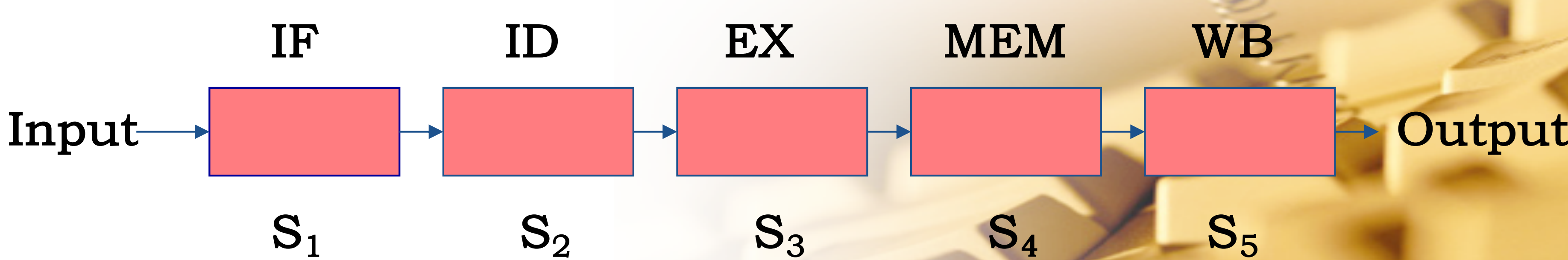
2. One-stage Overlap

IF_k	ID_k	EX_k		
		IF_{k+1}	Id_{k+1}	Ex_{k+1}

Chapter4 Pipelining Technology

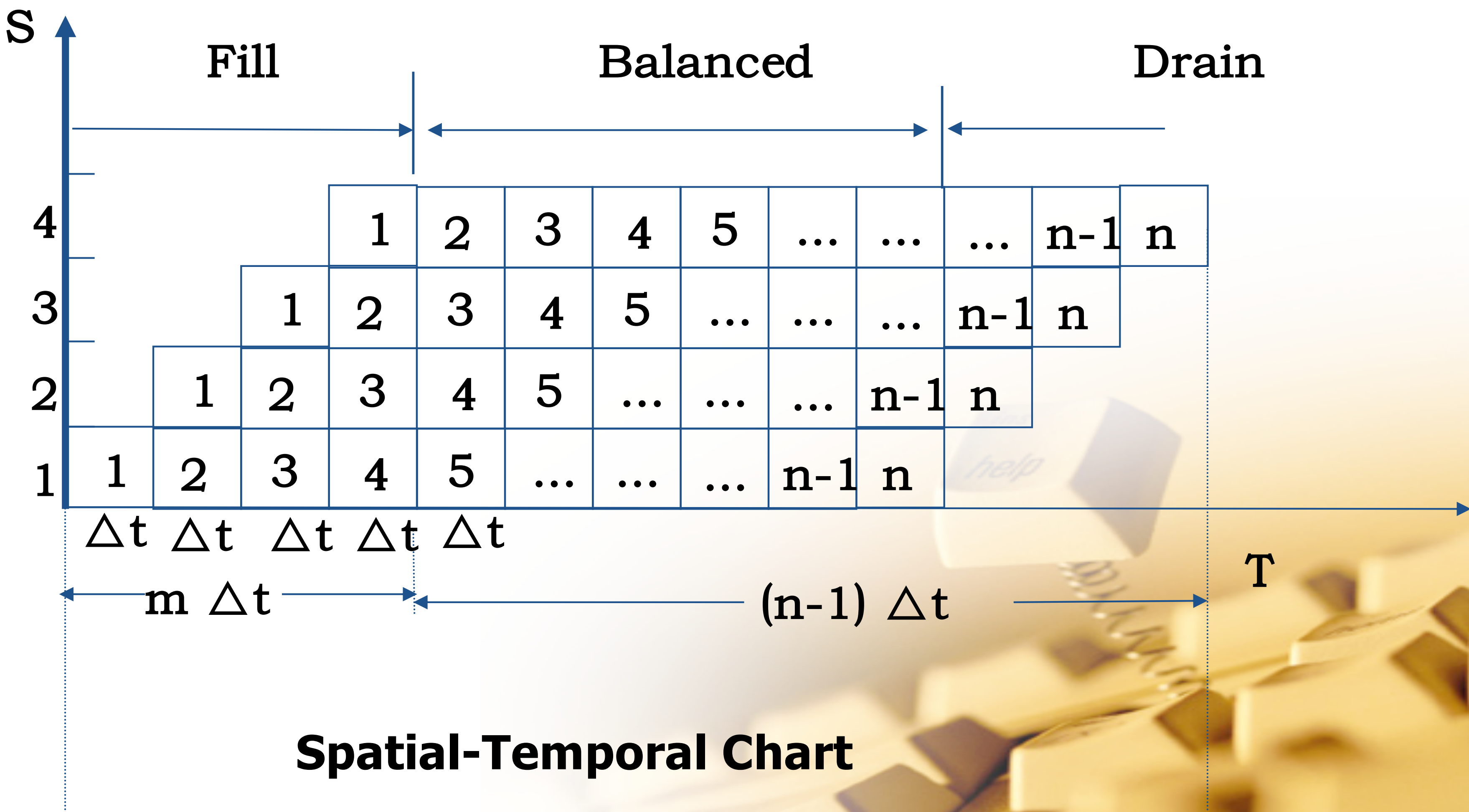


Pipelining Structure



Instruction Pipelining

Chapter4 Pipelining Technology



Spatial-Temporal Chart

Pipelining Classification

1. The Level of Processing

- **Arithmetic Pipelining**
- **Instruction Pipelining**
- **Macro Pipelining**

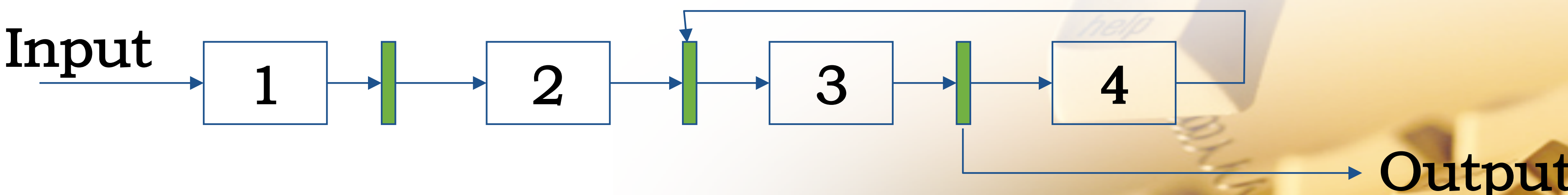
Chapter4 Pipelining Technology

2. Pipelining Function

- Unifunction Pipelining
- Multifunction Pipelining
  - ✓ Static Pipelining
  - ✓ Dynamic Pipelining

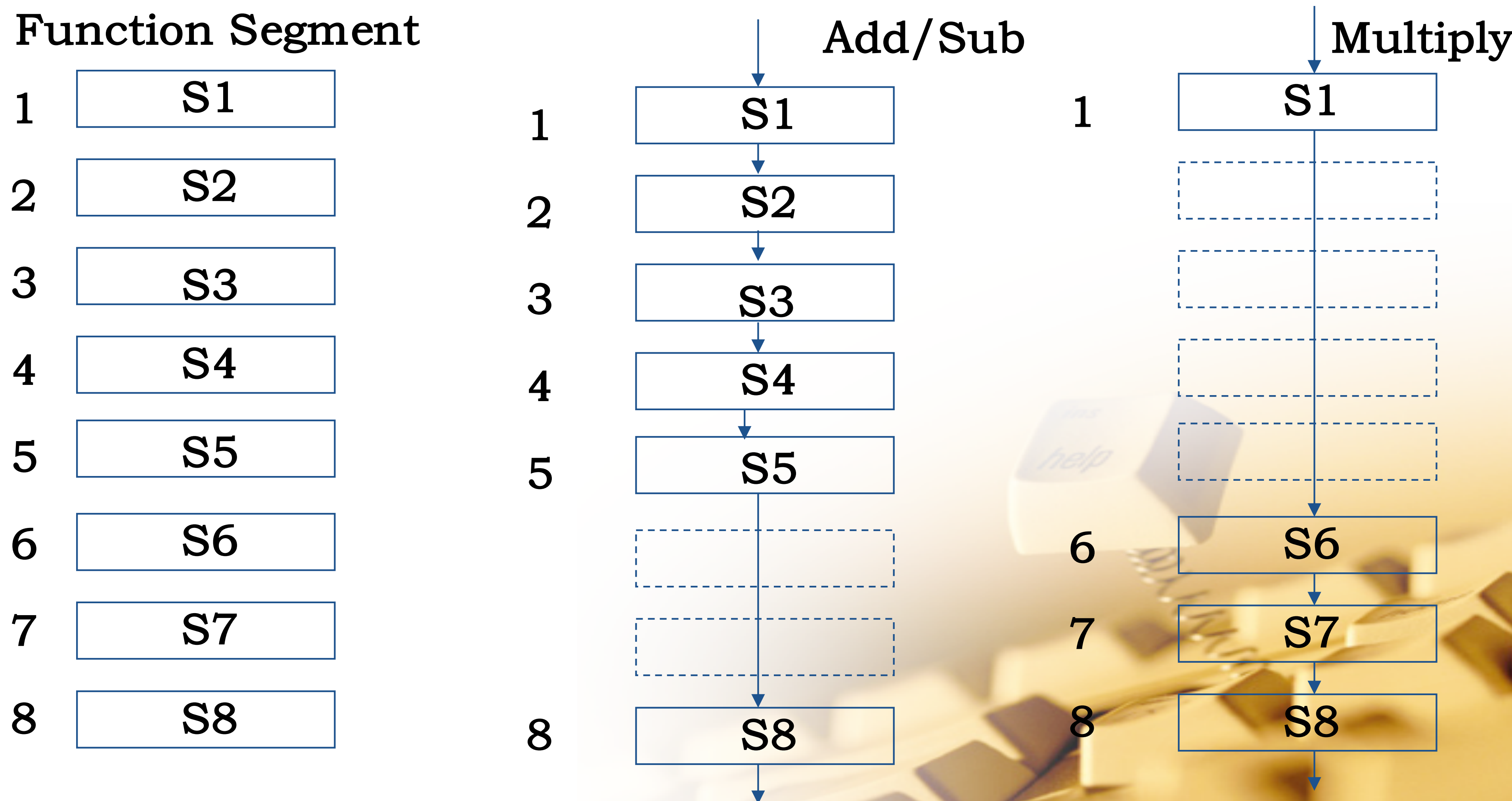
3. Connection Mode

- Linear Pipelining
- Non-Linear Pipelining

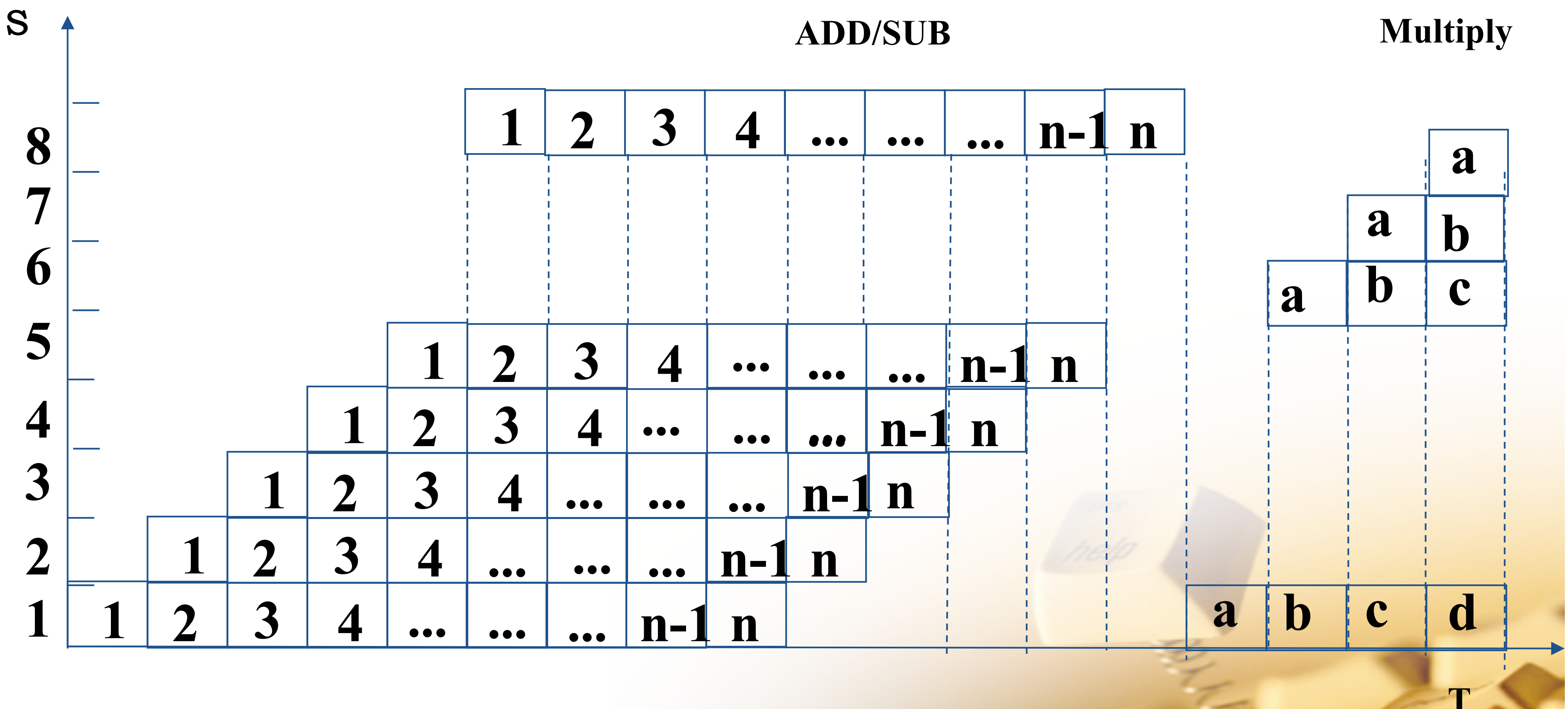


Non-Linear Pipelining

Pipelining Example

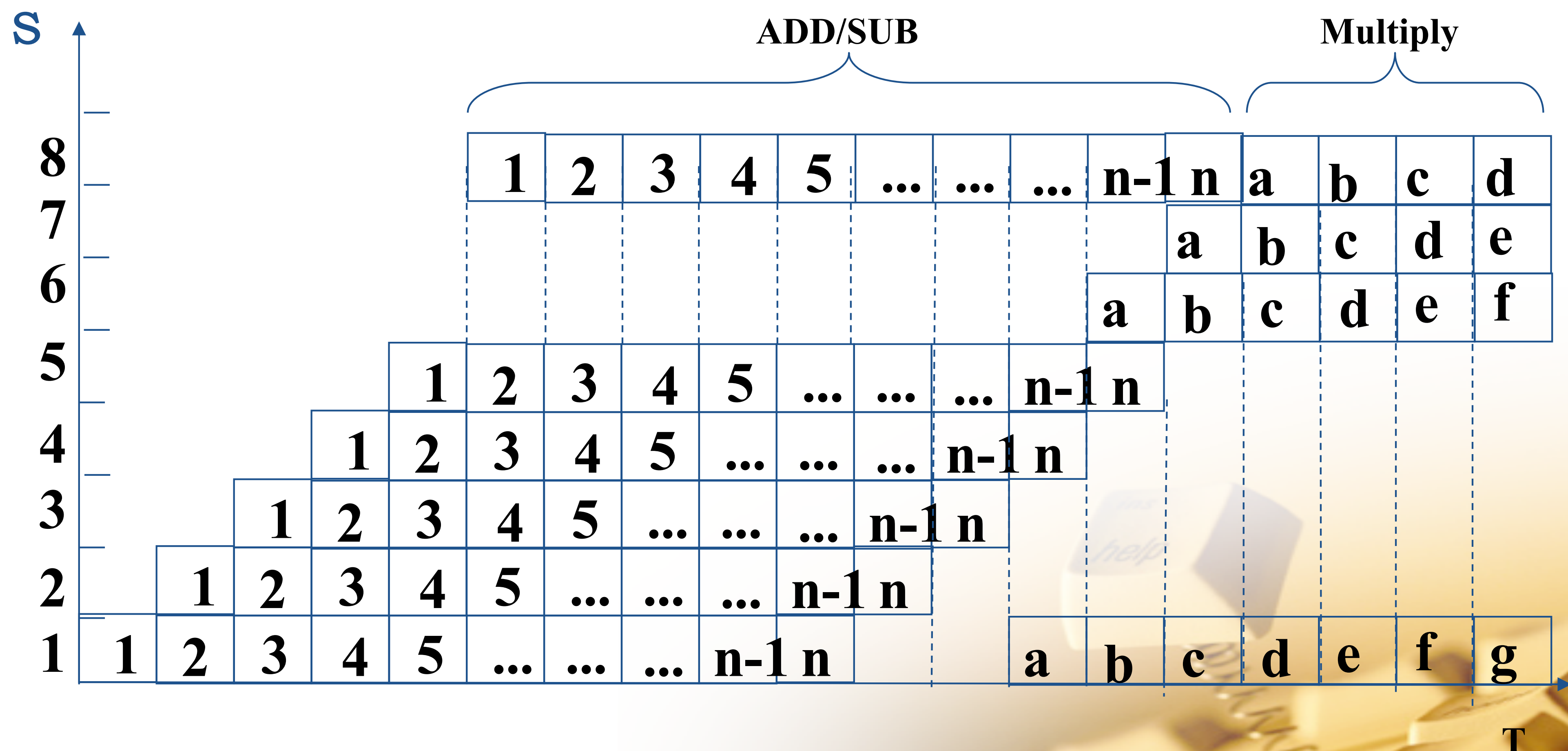


TI-ASC Multifunction Pipelining



Static Multifunction Pipelining

Chapter4 Pipelining Technology



Dynamic Multifunction Pipelining

Pipelining Performance

1. Performance Indicators

(1) TP(Throughput Rate):

The count of tasks can output from Pipelining per unit time.

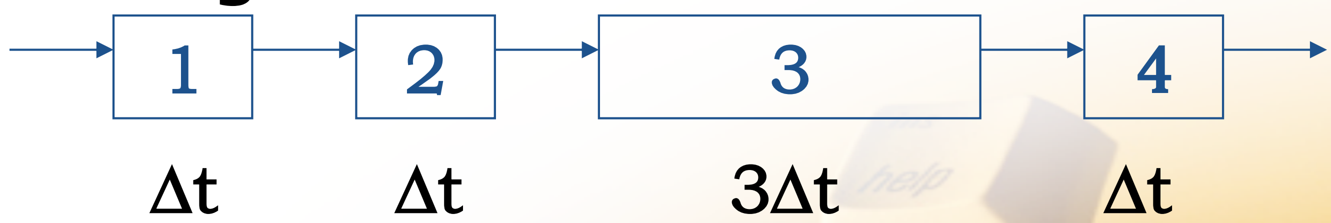
Chapter4 Pipelining Technology

- **Maximum Throughput Rate**：

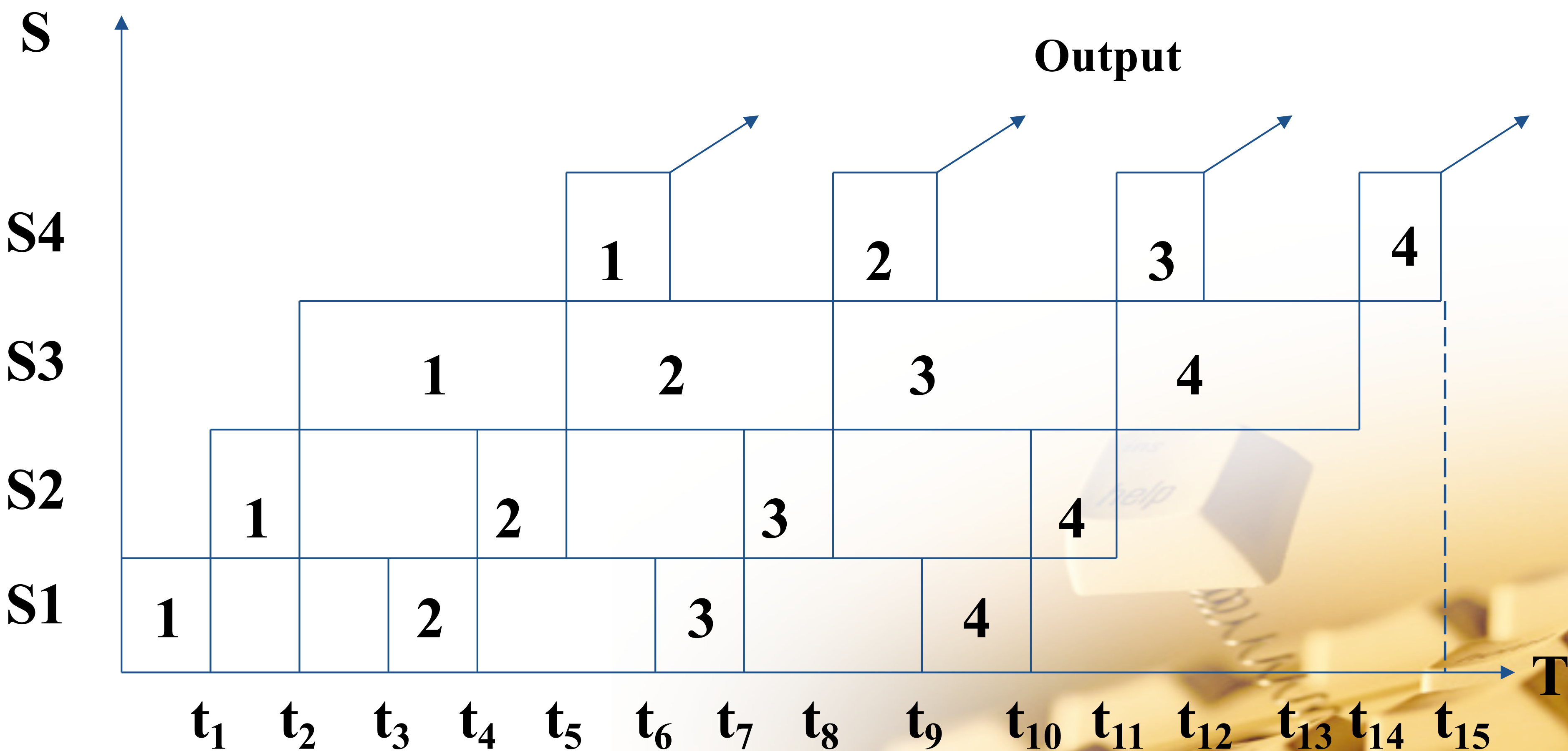
  $TP_{max}=1/\Delta t$ (ideal condition)

  $TP_{max}=1/\max\{\Delta t_1,\ \Delta t_2,\ \Delta t_3,\ \Delta t_4\}$

- **Bottleneck Segment**

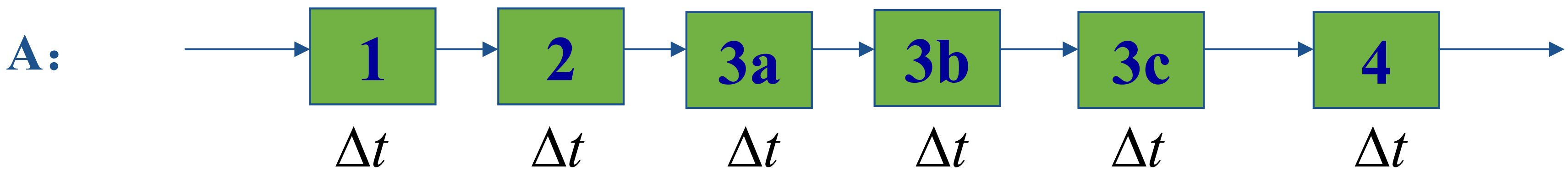


Chapter4 Pipelining Technology

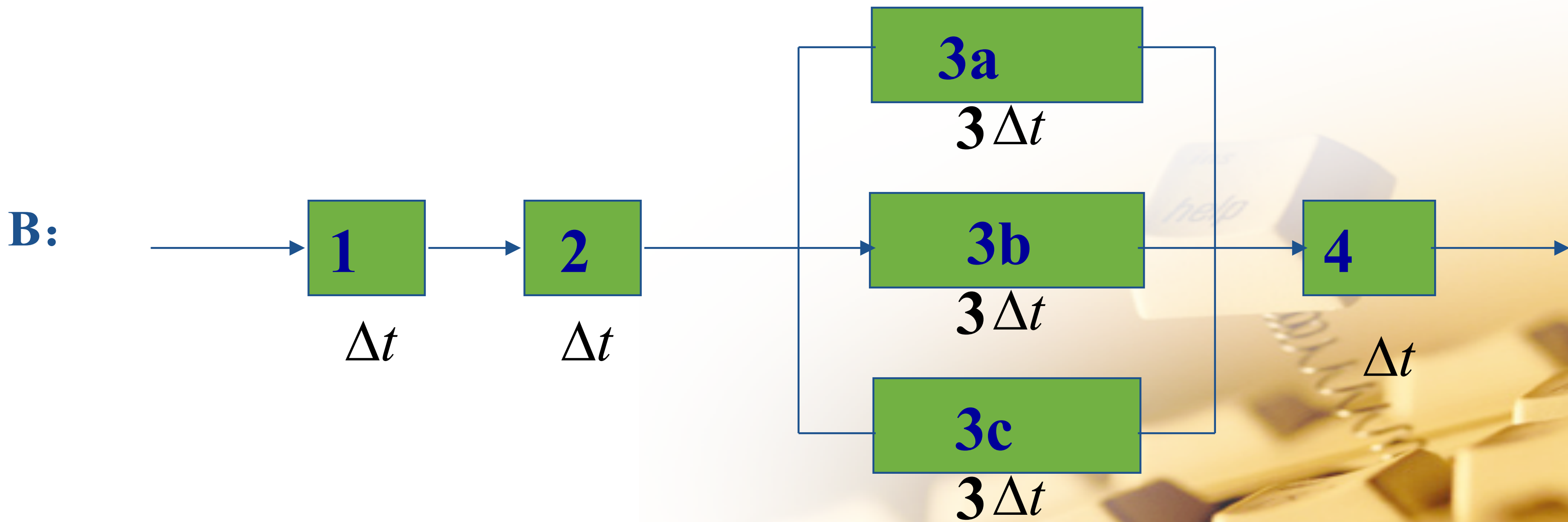


TPmax=1/3Δt

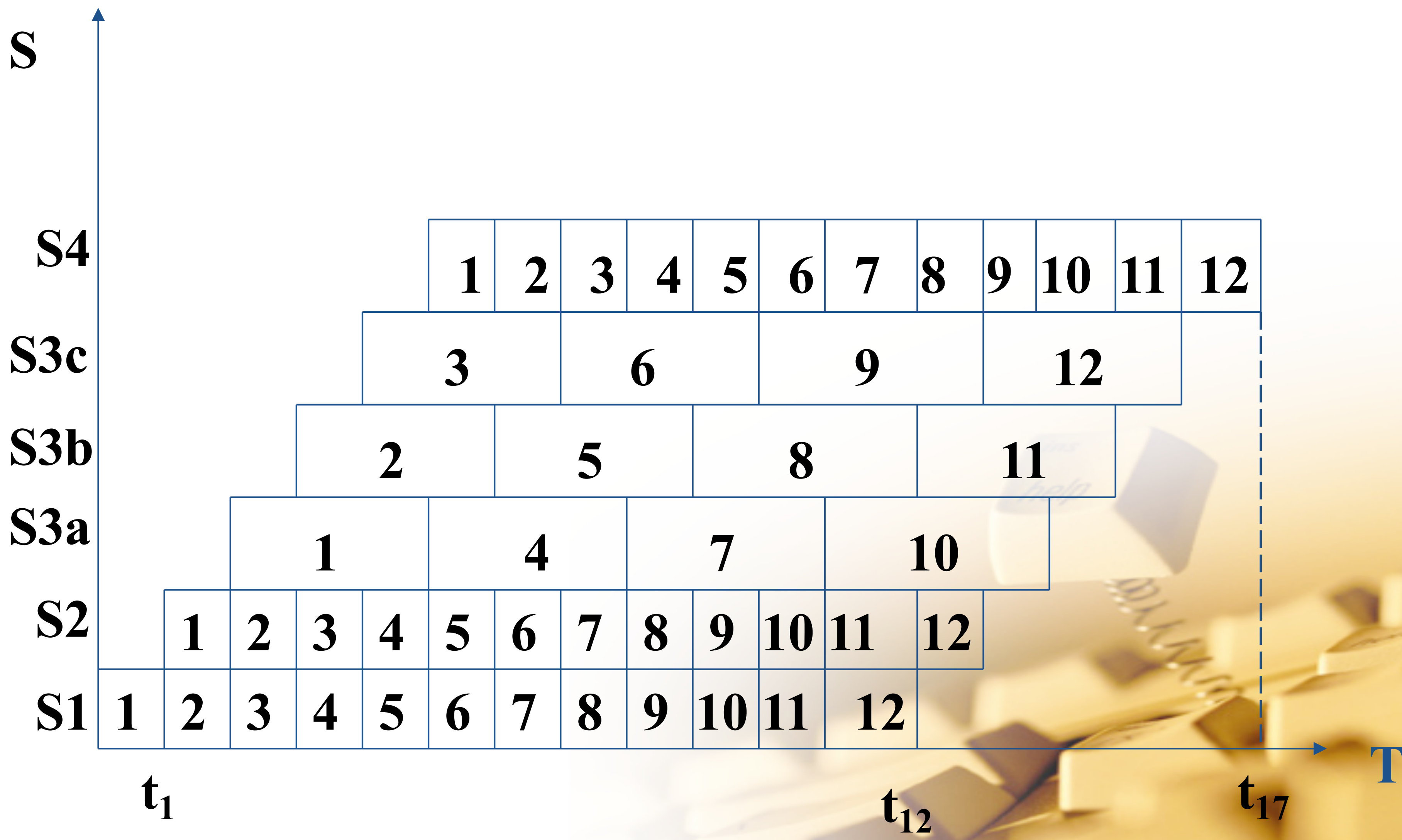
Chapter4 Pipelining Technology



Bottleneck Segmentation

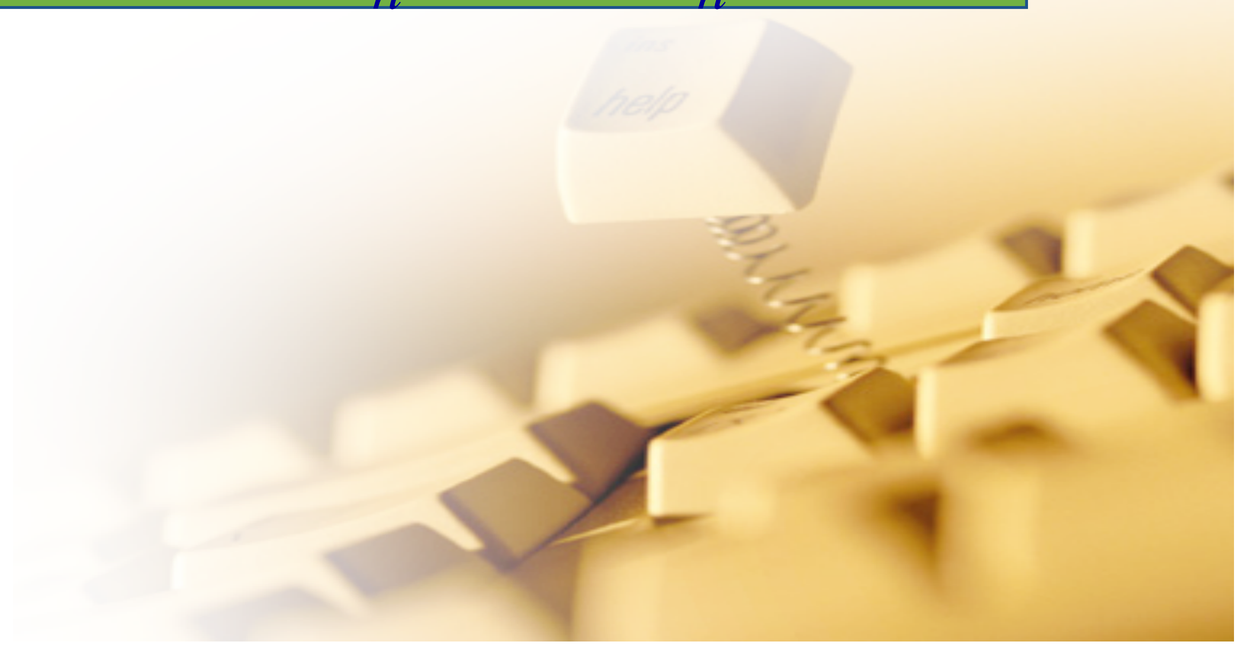


Bottleneck Replication


S
S4
1 2 3 4 5 6 7 8 9 10 11 12
S3c
3 6 9 12
S3b
2 5 8 11
S3a
1 4 7 10
S2
1 2 3 4 5 6 7 8 9 10 11 12
S1
1 2 3 4 5 6 7 8 9 10 11 12
T
t_1
t_{12}
t_{17}

✓Actual Throughput Rate

$$TP = \frac{n}{m\Delta t_0 + (n-1)\Delta t_0} = \frac{1}{\Delta t_0 (1 + \frac{m-1}{n})} = \frac{TP\max}{1 + \frac{m-1}{n}}$$



(2) Efficiency

Each Segment Time is Equal

$$\eta = \frac{m \cdot n \cdot \Delta t_0}{m \cdot T} = \frac{n \cdot \Delta t_0}{m\Delta t_0 + (n-1)\,\Delta t_0}$$

Each Segment Time is Not Equal

$$\eta = \frac{n * \sum_{i=1}^{m} \Delta t_i}{m * \left[\sum_{i=1}^{m} \Delta t_i + (n-1)\Delta t_j\right]}$$

Δt_j is the processing time of bottleneck segment

(3) Speedup Ratio

Each Segment Time is Equal

$$Sp=\frac{T_{non\text{-}Pipelining}}{T_{Pipelining}}=\frac{n*m*\Delta t}{m*\Delta t+(n-1)*\Delta t}=\frac{n*m}{m+n-1}=\frac{m}{1+\frac{m-1}{n}}$$

Each Segment Time is Not Equal

$$Sp=\frac{n*\sum_{i=1}^{m}\Delta ti}{\sum_{i=1}^{m}\Delta ti+(n-1)*\Delta tj}$$

Ex1: Four-stages pipelining, $\triangle t_1=\triangle t_3=\triangle t_4=\triangle t$, $\triangle t_2=3\triangle t$,Please write out TP, η and SP when the task count(n) is 4 or 10.

(1) Analysis Method: Assume n =10

$$TP=\frac{n}{\sum_{i=1}^{m}\triangle ti+(n-1)\triangle tj}$$

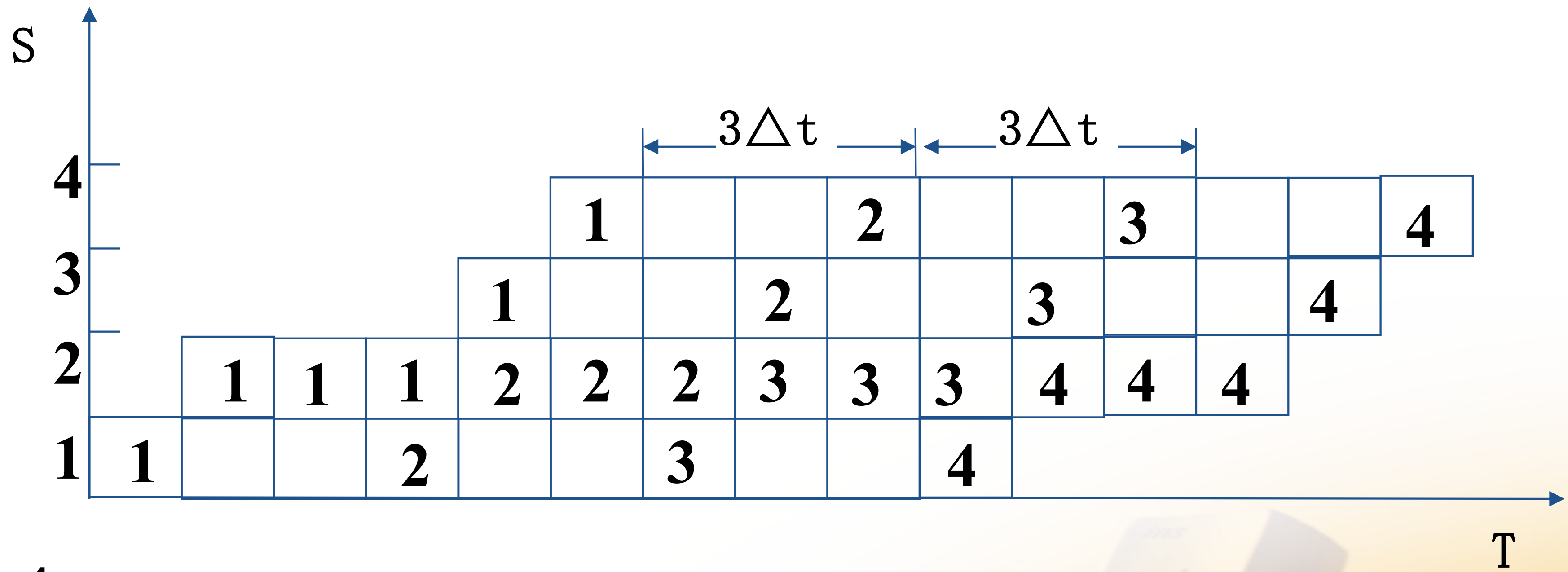
Chapter4 Pipelining Technology

$$Sp=\frac{n*\sum_{i=1}^{m}\triangle ti}{\sum_{i=1}^{m}\triangle ti+(n-1)*\triangle tj}=\frac{10*6\triangle t}{(6+3*9)\triangle t}=\frac{20}{11}=1.8$$

$$TP=\frac{10}{6*\triangle t+3*9*\triangle t}=\frac{10}{33*\triangle t}=0.303/\triangle t$$

$$\eta=\frac{6*10\triangle t}{4*6\triangle t+9*3*4\triangle t}=\frac{60}{24+108}=\frac{5}{11}\approx 45\%$$

(2) Spatial-Temporal Chart Method



n=4：

TP=4/((6+3*3) △t)=4/(15△t)=0.267/△t

η=6*4 △t /(4*15△t)=2/5=40%

Sp=4*6△t /15△t=8/5=1.6

m=4

	TP(1 / $\triangle$t)	η	Sp
n=4	0.267	40%	1.6
n=10	0.303	45%	1.8
n=100	0.33	50%	1.98

Chapter4 Pipelining Technology

Ex2: Assume that there is a four-stages floating-point addition pipelining and every stage has the same processing time. If we use this Pipelining to calculate the following formula

$$Z=A+B+C+D+E+F+G+H$$

Please write out TP、η and Sp.

S																
S4				1	2	3	4		5		6				7	
S3			1	2	3	4		5		6				7		
S2		1	2	3	4		5		6				7			
S1	1	2	3	4		5		6				7				

T

Solution:

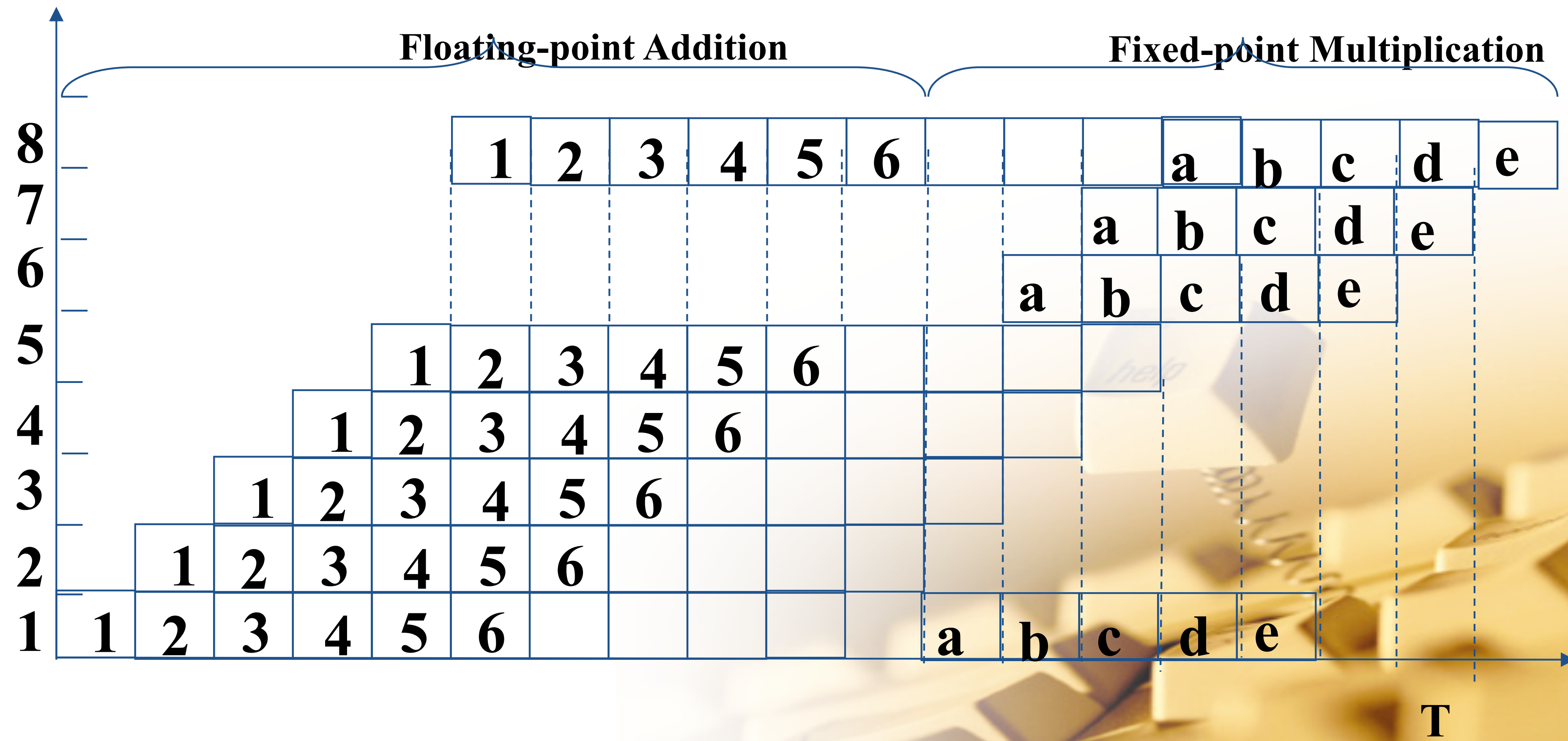
TP=7/（15△t） = 0.47/△t

η =7*4/(15*4)=7/15 = 47%

Sp=4*7/15=28/15 = 1.87

$$Z=\underbrace{\underbrace{\underbrace{A+B}_{1}+\underbrace{C+D}_{2}}_{5}+\underbrace{\underbrace{E+F}_{3}+\underbrace{G+H}_{4}}_{6}}_{7}$$

Ex3. Please write out TP、η and Sp when we use the TI-ASC multifunction pipelining to calculate 6 floating-point addition and 5 fixed-point （m=8,n=11）



Chapter4 Pipelining Technology

$T_{add}=（6+(6-1)）*\Delta t = 11\Delta t$

$T_{mul}=（4+(5-1)）*\Delta t = 8\Delta t$

$T=T_{add}+T_{mul}= 19\Delta t$

$TP = 11/(（11+8）\Delta t）=11/19\Delta t=0.58/\Delta t$

$\eta =（6*6+5*4）\Delta t /(19*8\Delta t)=6/52=7/19=36.8\%$

$Sp =（6*6+5*4）\Delta t /19\Delta t = 56\Delta t /19\Delta t =2.95$

Chapter4 Pipelining Technology

Ex4: Three-stages pipelining, $\triangle t_1 = \triangle t_3 = \triangle t$, $\triangle t_2 = 3\triangle t$.

(1)Please write out Tp and η when the task count(n) is 3 or 30.

(2)Please try to eliminate the bottleneck problem and write out Tp and η when the task count(n) is 3 or 30.

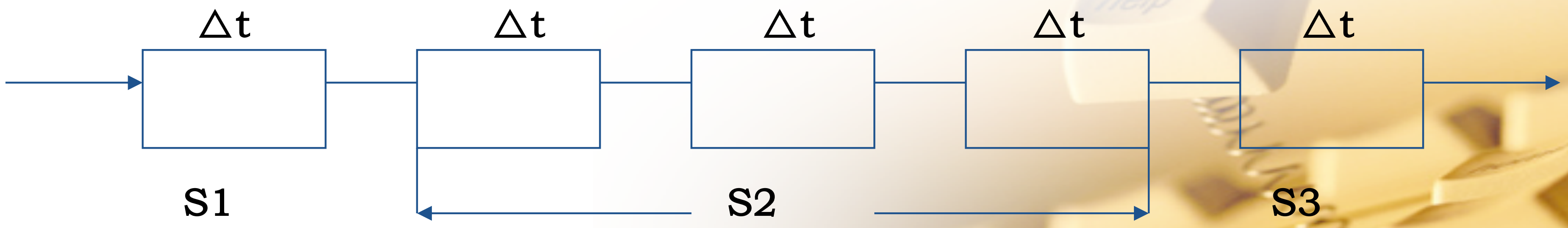
Solution:

(1) $n=3$, $m=3$, $\triangle t_1=\triangle t$, $\triangle t_2=3\triangle t$, $\triangle t_3=\triangle t$, $\triangle t_j=3\triangle t$

$$TP=\frac{n}{\sum_{i=1}^{m}\Delta t_i+(n-1)\Delta t_j}=\frac{3}{5\Delta t+2*3\Delta t}=\frac{3}{11\Delta t}$$

$$\eta = \frac{n\sum_{i=1}^{m}\Delta t_i}{m\left[\sum_{i=1}^{m}\Delta t_i + (n-1)\Delta t_j\right]} = \frac{3\times 5\Delta t}{3\times 11\Delta t} = \frac{5}{11} = 45\%$$

(2) Bottleneck Segmentation



Chapter4 Pipelining Technology

n=3，m=5，$\Delta t_i = \Delta t_j = \Delta t$

$$TP = \frac{3}{\sum_{i=1}^{5} \Delta t_i + (3-1)\Delta t_i} = \frac{3}{5\Delta t + 2\Delta t} = \frac{3}{7\Delta t} = 0.43/\Delta t$$

$$\eta = \frac{3 \bullet \sum_{i=1}^{5} \Delta t_i}{5 * 7\Delta t} = \frac{3 * 5\Delta t}{5 * 7\Delta t} = \frac{3}{7} = 43\%$$

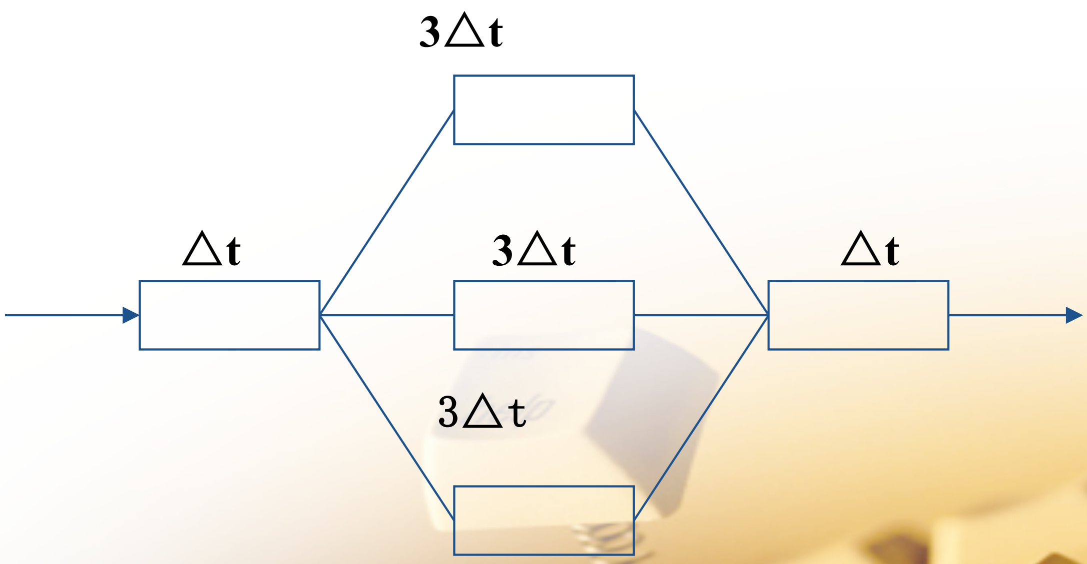
n=30：

TP=0.88/Δt

η=88%

Chapter4 Pipelining Technology

Bottleneck	No Bottleneck
n=3:	n=3:
TP=0.237/Δt	TP=0.43/Δt
η=45%	η=43%
n=30:	n=30:
TP=0.326/Δt	TP=0.88/Δ t
η=54%	η=88%



Bottleneck Replication

Ex5. Assume that a dual-function static pipelining can execute addition and multiplication and include 4 stages. Every stage's processing time is separately △t、2△t、2△t、3△t. The connection of addition is 1→2→4, the connection of multiplication is 1→3→4. Now calculate the following formula

A*（B+C*（D+E*F））+G*H

(1) Please adjust the order of calculation and draw the spatial-temporal chart which can get the optimal throughput and write out the total calculation time and η.

(2) If we want to eliminate the bottleneck problem by bottleneck segmentation, how long we can finish the calculation?

(3) If we want to eliminate the bottleneck problem by bottleneck replication, please write out the total calculation time and η.

Chapter4 Pipelining Technology

$$\underbrace{\underbrace{\underbrace{A*B}_{4} + \underbrace{\underbrace{A*C}_{1}*D}_{5}}_{7} + \underbrace{\underbrace{\underbrace{A*C}_{1} * \underbrace{E*F}_{2}}_{6} + \underbrace{G*H}_{3}}_{8}}_{9}$$

（1）$T=36\,\Delta t$，$\eta=38\%$

（2） $T=26\,\Delta t$

（3） $\eta=26\%$

Pipelining Hazard

Hazard is a condition that prevents an instruction in the Pipelining from executing its next scheduled pipelining stage.

- **Structural hazards**

  These are conflicts over hardware resources.

- **Data hazards**

  Instruction depends on result of prior computation which is not ready (computed or stored) yet.

- **Control hazards**

  Branch condition and the branch PC are not available in time to fetch an instruction on the next clock.

Structural Hazards

Ex1:

$$X_3 = X_2 * X_1$$

$$X_6 = X_4 * X_5$$

Ex2:

T / I	1	2	3	4	5	6	7	8
Load	IF	ID	EX	MEM	WB			
i+1		IF	ID	EX	MEM	WB		
i+2			IF	ID	EX	MEM	WB	
i+3				IF	ID	EX	MEM	WB
i+4					IF	ID	EX	MEM

Conflict

Solution:

T / I	1	2	3	4	5	6	7	8	9
Load	IF	ID	EX	MEM	WB				
i+1		IF	ID	EX	MEM	WB			
i+2			IF	ID	EX	MEM	WB		
i+3				Stall	IF	ID	EX	MEM	WB
i+4						IF	ID	EX	MEM

Data Hazards

- ✓ **RAW(Read After Write) (WR)**

  i: R1+R2->R3 j: R3*R4->R5

- ✓ **WAR(Write After Read) (RW)**

  i: R3*R4->R5 j: R1+R2->R3

- ✓ **WAW(Write After Write) (WW)**

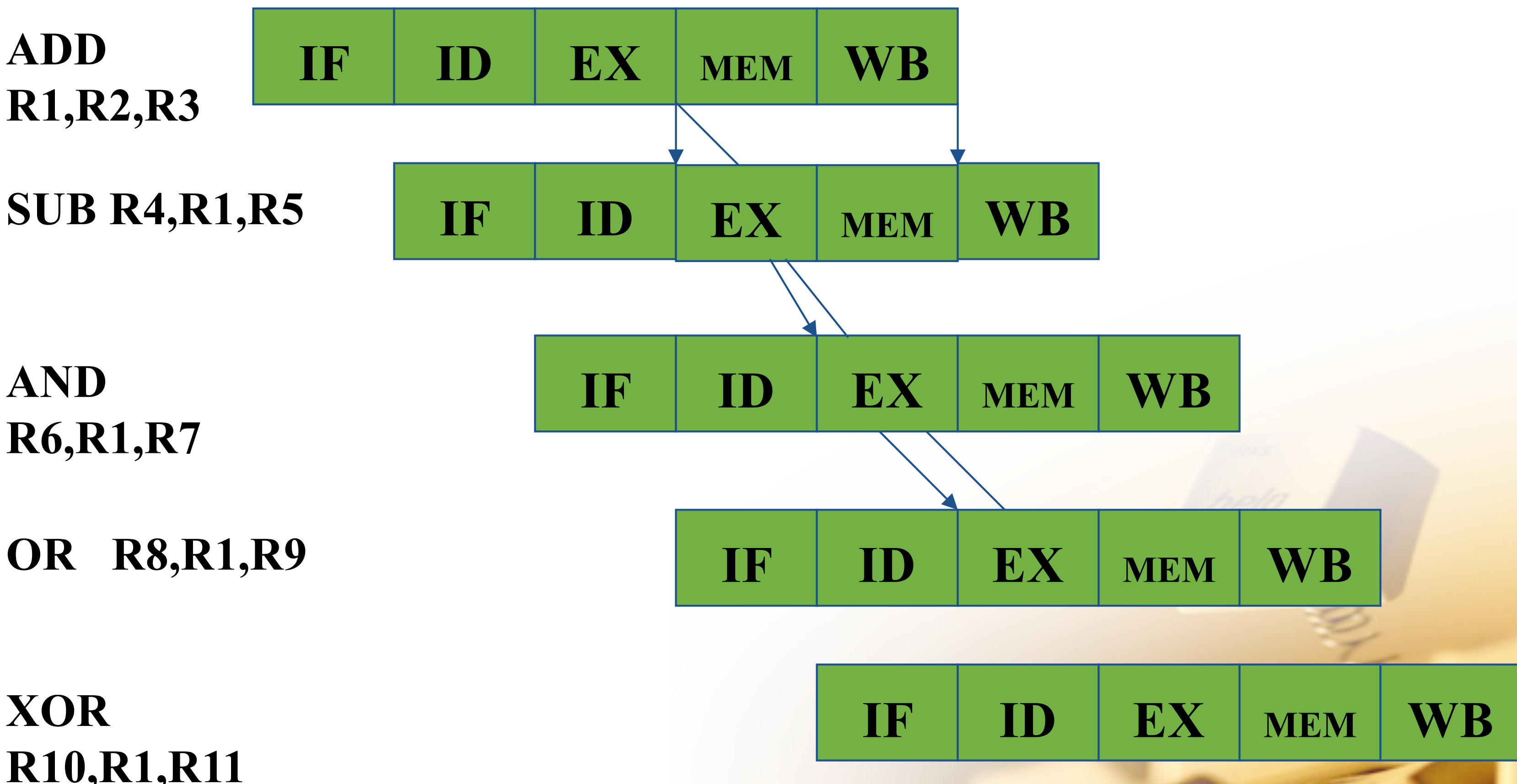
  i: R1*R2->R3 j: R4+R5->R3

How to deal with data hazards？

- **Insert Stalls**
- **Forwarding**

- **Instead of waiting to store the result, we forward it immediately to the instruction that wants it.**
- **Mechanically, we add buses to the data path to move these values.**
- **These bused always "point backwards" in the data path, from later stages to earlier stages.**

Chapter4 Pipelining Technology



Forwarding

Chapter4 Pipelining Technology

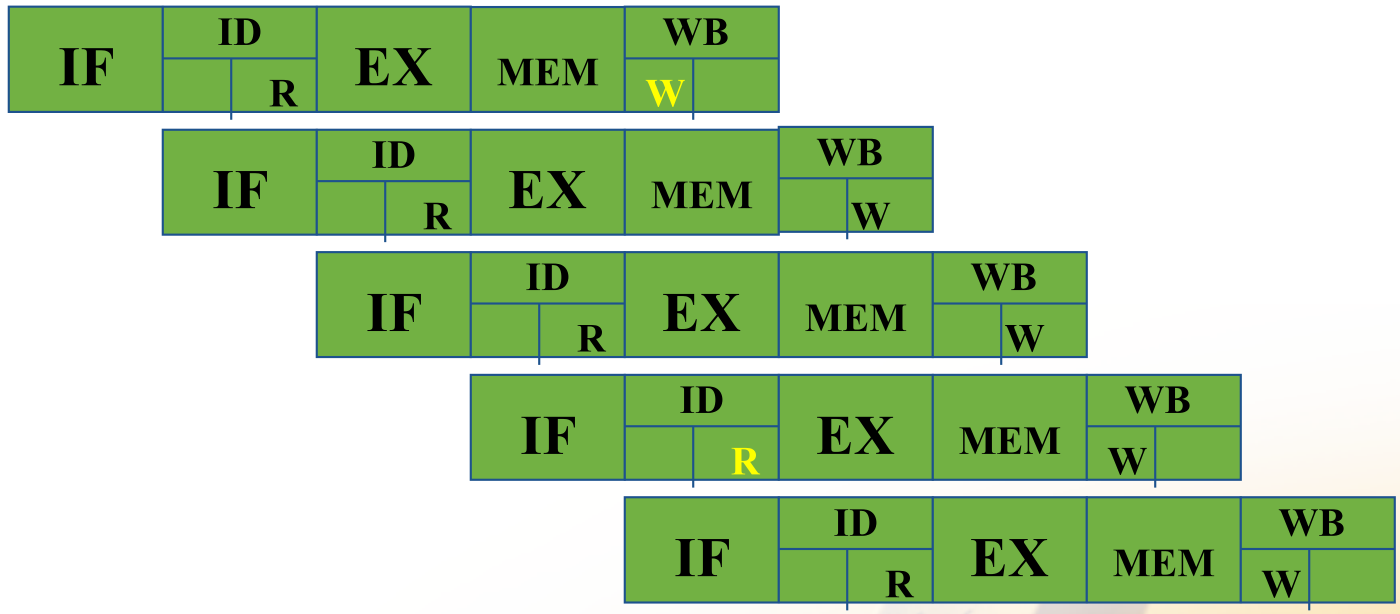
ADD R1,R2,R3

SUB R4,R1,R5

AND R6,R1,R7

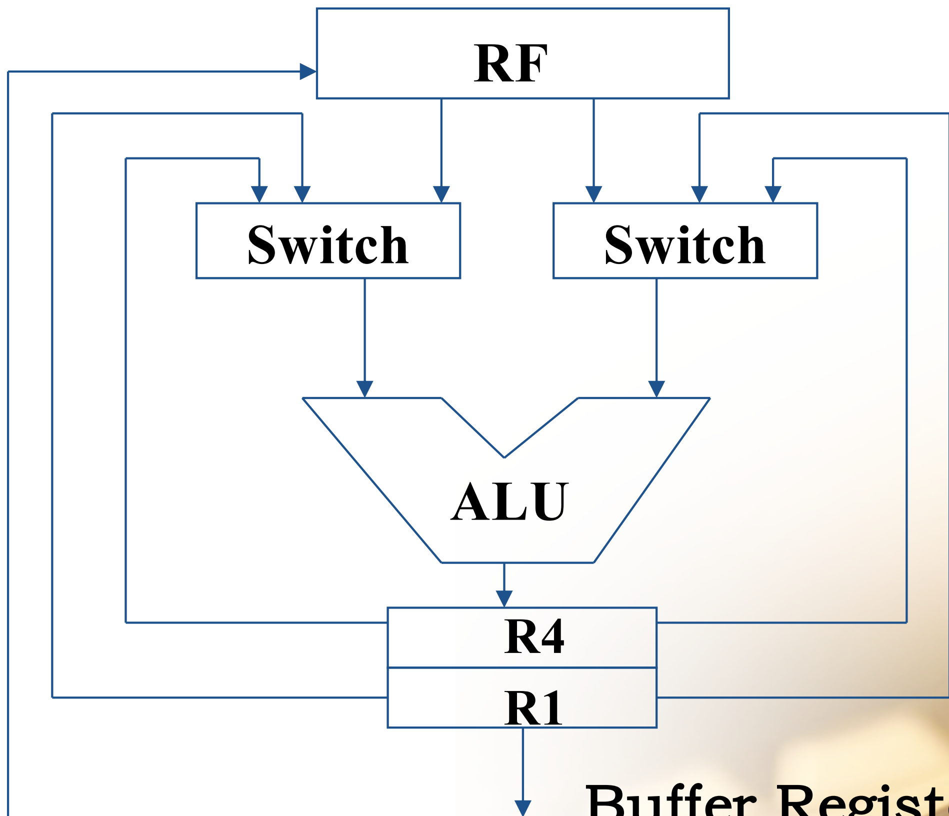
OR R8,R1,R9

XOR R10,R1,R11



Reduce Forwarding Count

Chapter4 Pipelining Technology

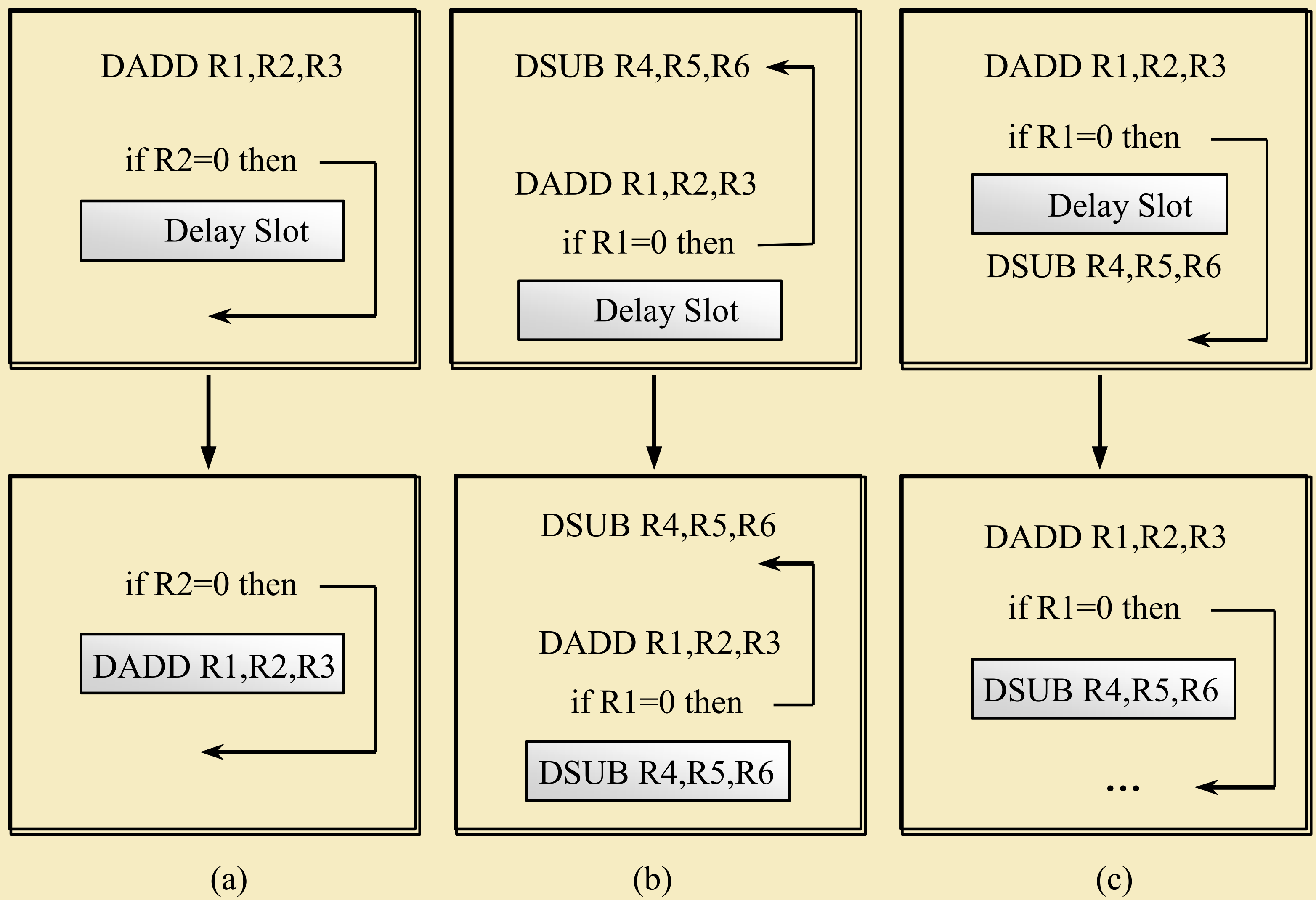


Buffer Register (R1, R4)

Chapter4 Pipelining Technology

- **Control Hazards**
  - **Form condition code in advance**
  - **Static branch prediction**
  - **Assume branch not taken or taken.**
  - **Prefetching branch target.**
  - **Speed up short loop processing.**
  - **Dynamic branch prediction**
  - **Delayed Branch**

Delayed Branch



Chapter4 Pipelining Technology

Pipelining Interrupt Handling

- ✓ **Imprecise Breakpoint**
- ✓ **Precise Breakpoint**

Non-linear Pipelining Conflict and Scheduling

Non-linear Pipelining Conflict：

Function Segment Conflict

The problem needs to be solved is how long next task can be taken into pipelining.

Reservation Table

S/t	1	2	3	4	5	6	7	8	9
1	X								X
2		X	X					X	
3				X					
4					X	X			
5							X	X	

Forbidden List

$$F=\{1,\ 5,\ 6,\ 8\}$$

Collision Vector

$$C=(C_n C_{n-1} \ldots C_2\ C_1)$$

Ex1: If $F=\{1,\ 3,\ 6\}$, then $n=6$

$C =\ (100101)$

Ex2: If $F=\{1,\ 5,\ 6,\ 8\}$, then $n=8$

$C =\ (10110001)$

How to generate new collision vector?

Ex1: If C=(10110001) and the second task enters into pipelining after 3△t

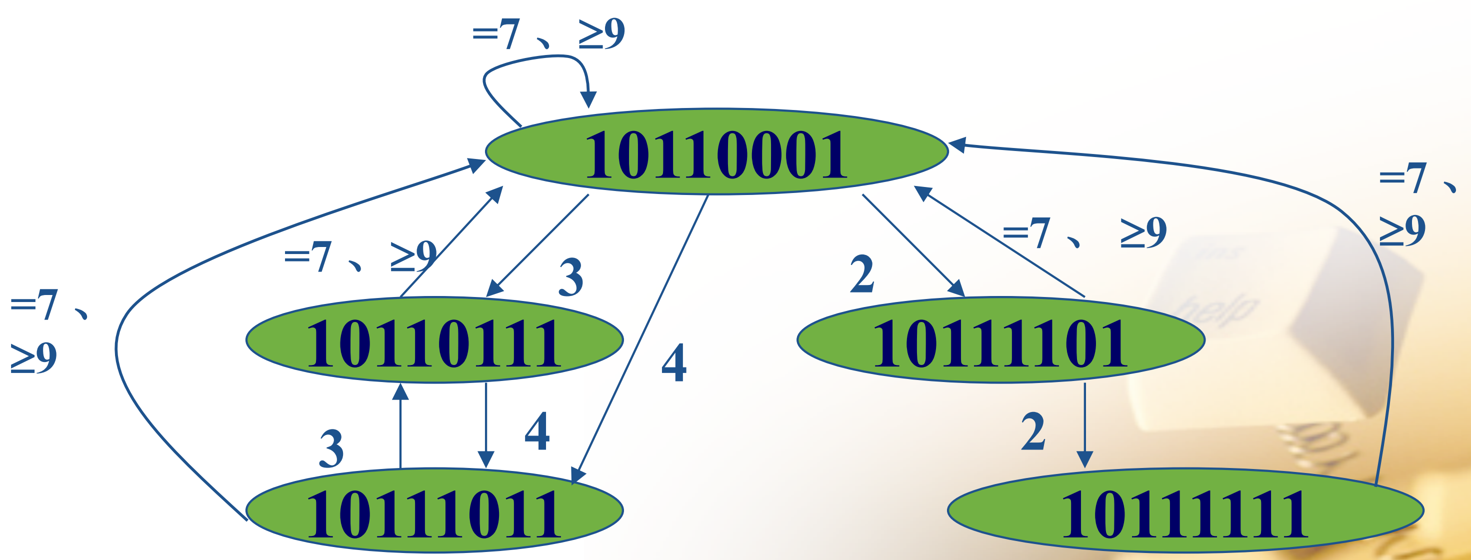
$$
\begin{array}{rll}
\rightarrow 3: & 00010110 & \\
\vee & 10110001 & \text{(Original)} \\
\hline
 & 10110111 & \text{(New)}
\end{array}
$$

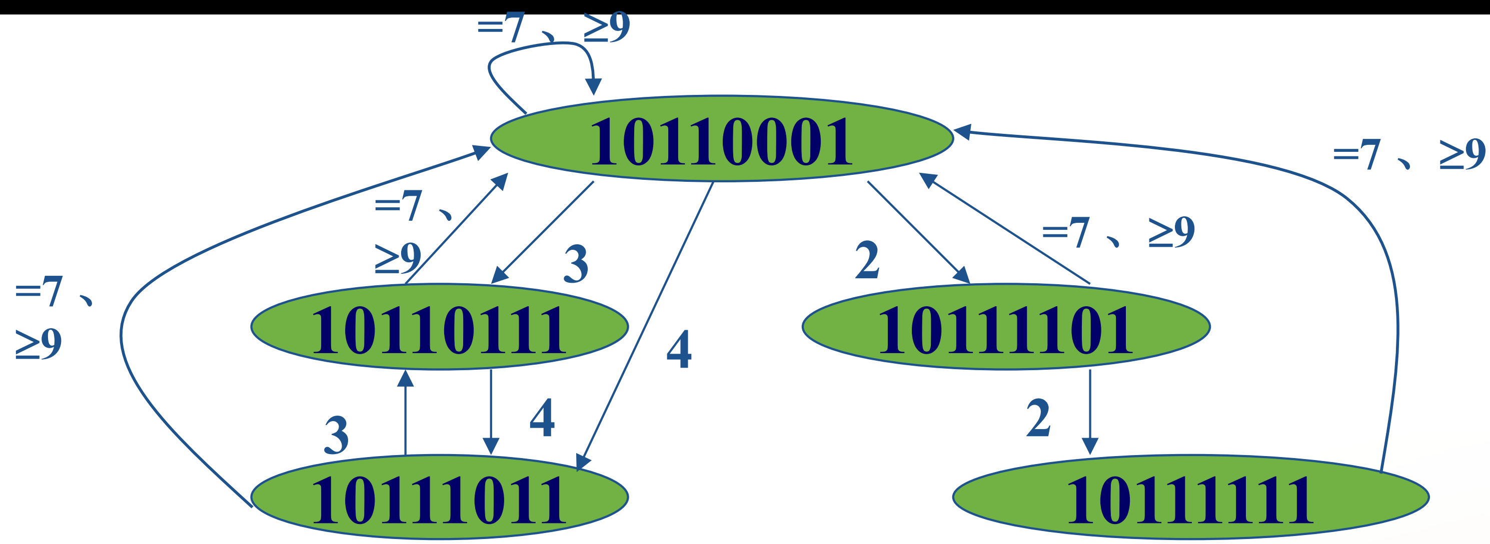
Ex2: If the third task enters into pipelining after 4△t

$$
\begin{array}{rll}
\rightarrow 4: & 00001011 & \\
\vee & 10110001 & \text{(Original)} \\
\hline
 & 10111011 & \text{(New)}
\end{array}
$$

State transition diagram

F={1，5，6，8}





Scheduling Policy		**Average Latency**
Different Interval	(2，2，7)	(2+2+7) / 3 = 3. 67
	(3，4)	(3+4) / 2 = 3. 5
Same Interval	(7)	7

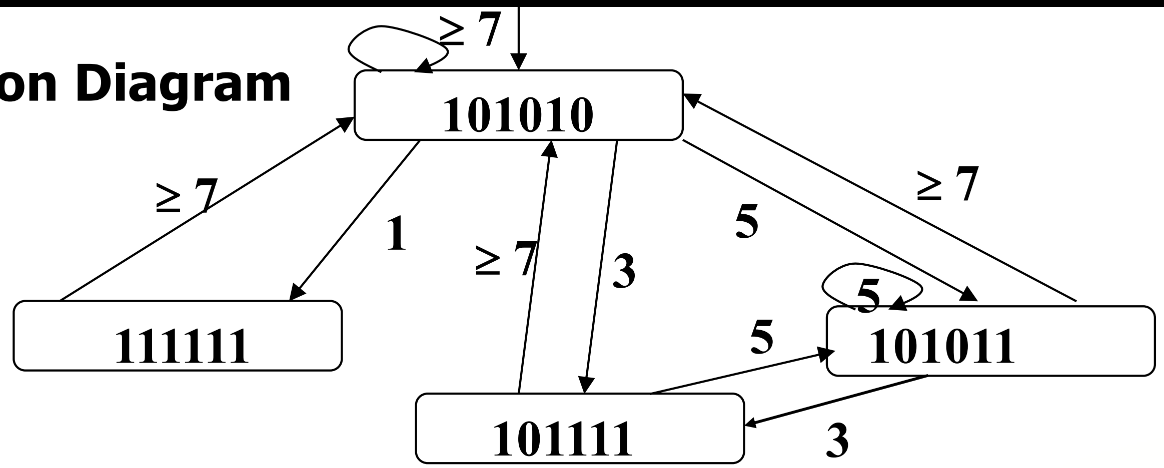
Example：**The following is the reservation table of a four-stages non-linear pipelining. Please write out the forbidden list(F) and the collision vector(C). Please draw the state transition diagram and get the minimum average latency, the maximum throughput and the best scheduling policy. If you use this scheduling policy to input 6 tasks, please write out the actual throughput.**

S/t	1	2	3	4	5	6	7
1	√				√		√
2		√		√			
3			√				
4				√		√	

Solution：**F={2，4，6}**

C=（101010）

State Transition Diagram



Scheduling Policy	Average Latency
(7)	7
(1, 7)	4
(3, 7)	5
(3, 5)	4
(5, 7)	6
(5)	5
(3, 5, 7)	5
(5, 3, 7)	5

From the above table we can get:

The minimum average latency is 4△t

TPmax=1/4△t

The optimal scheduling policy are (1,7), (3,5) and (5,3).

(1) If we use (1,7) policy to input 6 tasks, then

T=7+1+7+1+7+1=24 △t

TP=6/24 △t

(2) If we use (3,5) policy to input 6 tasks, then

$$T=7+3+5+3+5+3=26\,\triangle t$$

$$TP=6/26\,\triangle t$$

(3) If we use (5,3) policy to input 6 tasks, then

$$T=7+5+3+5+3+5=28\triangle t$$

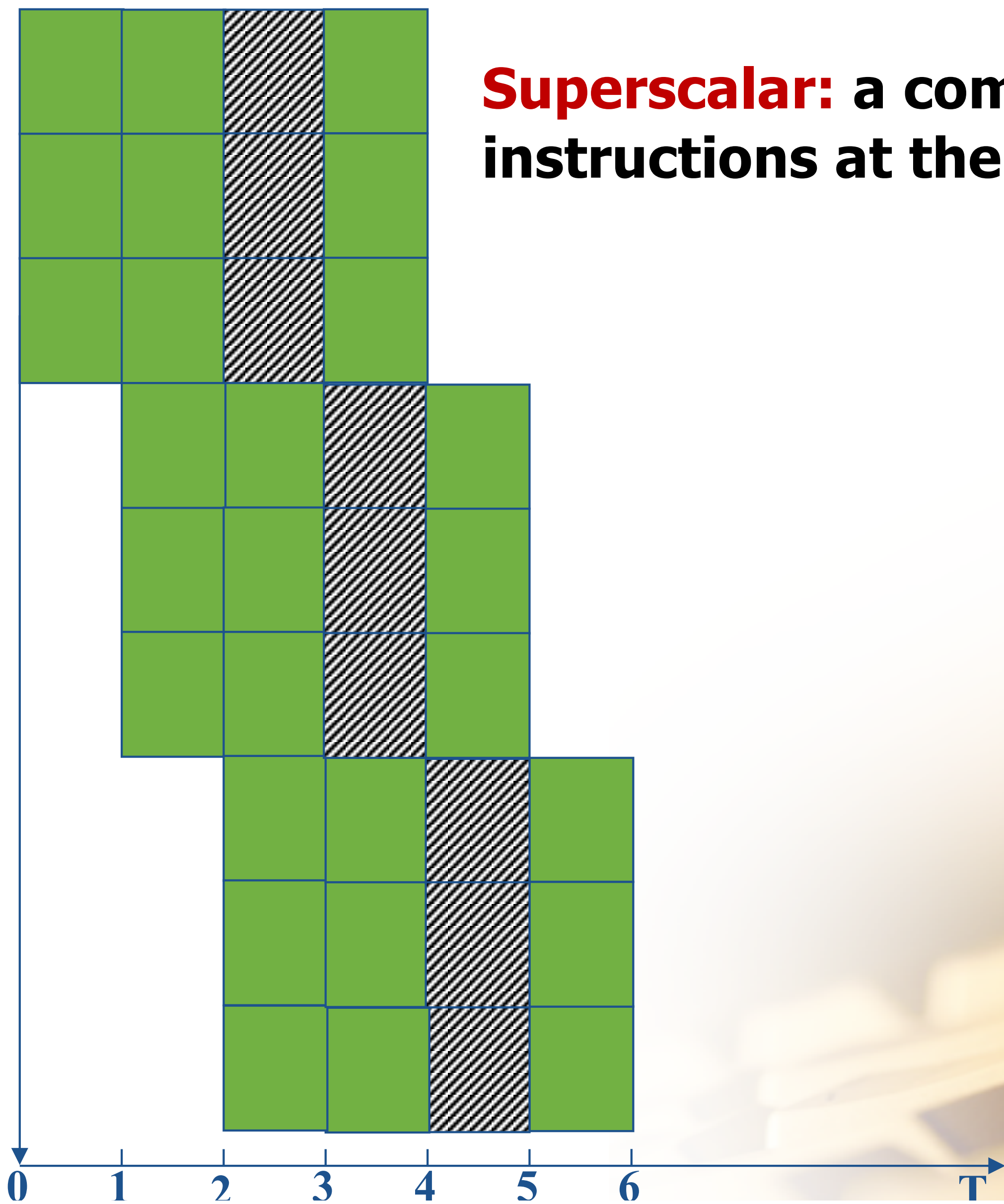
$$TP=6/28\,\triangle t$$

So we can get the best scheduling policy is (1,7) .

To Further Develop Instruction Parallelism

- **Superscalar**
- **Super Pipelining**
- **VLIW**

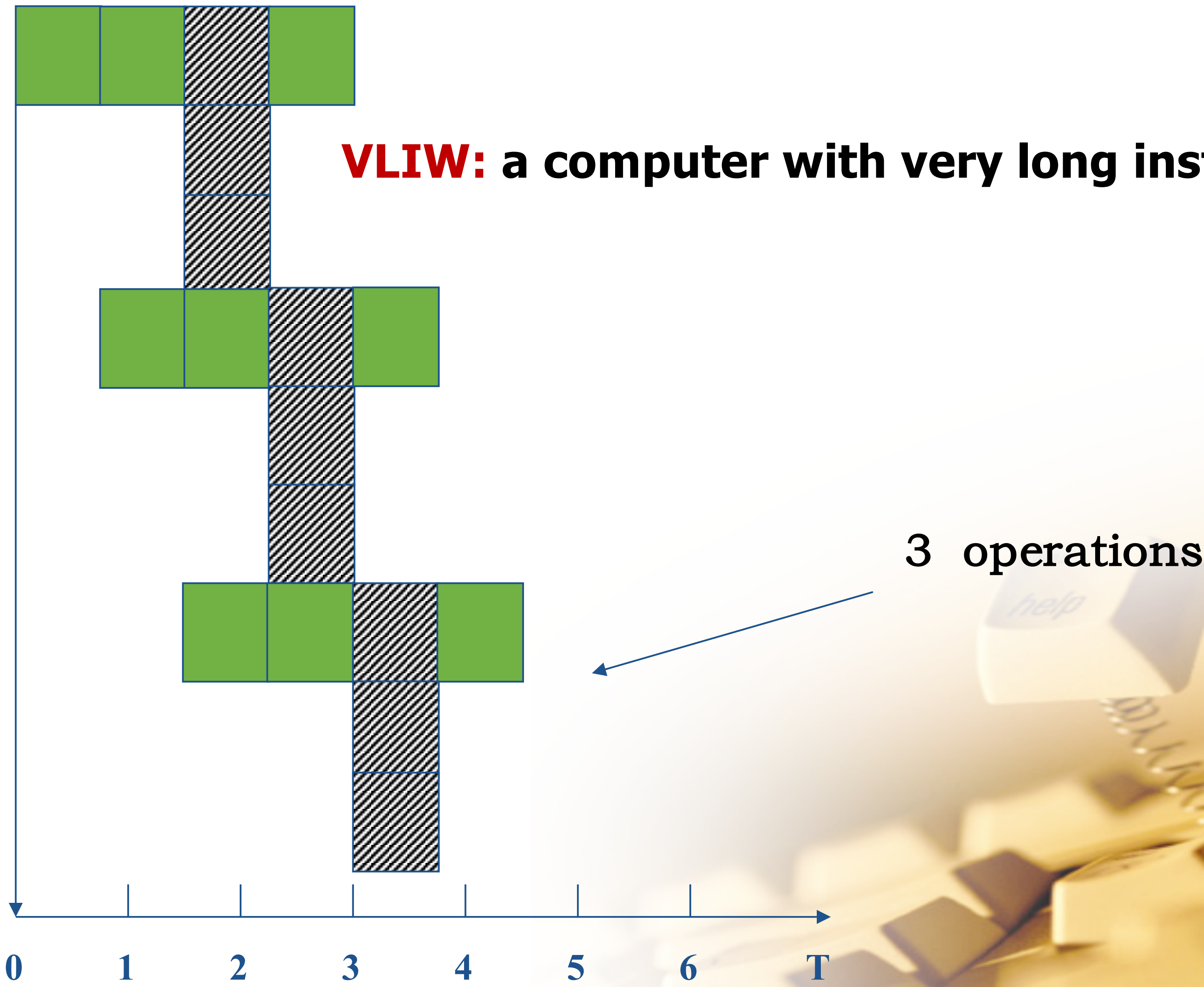
Chapter4 Pipelining Technology



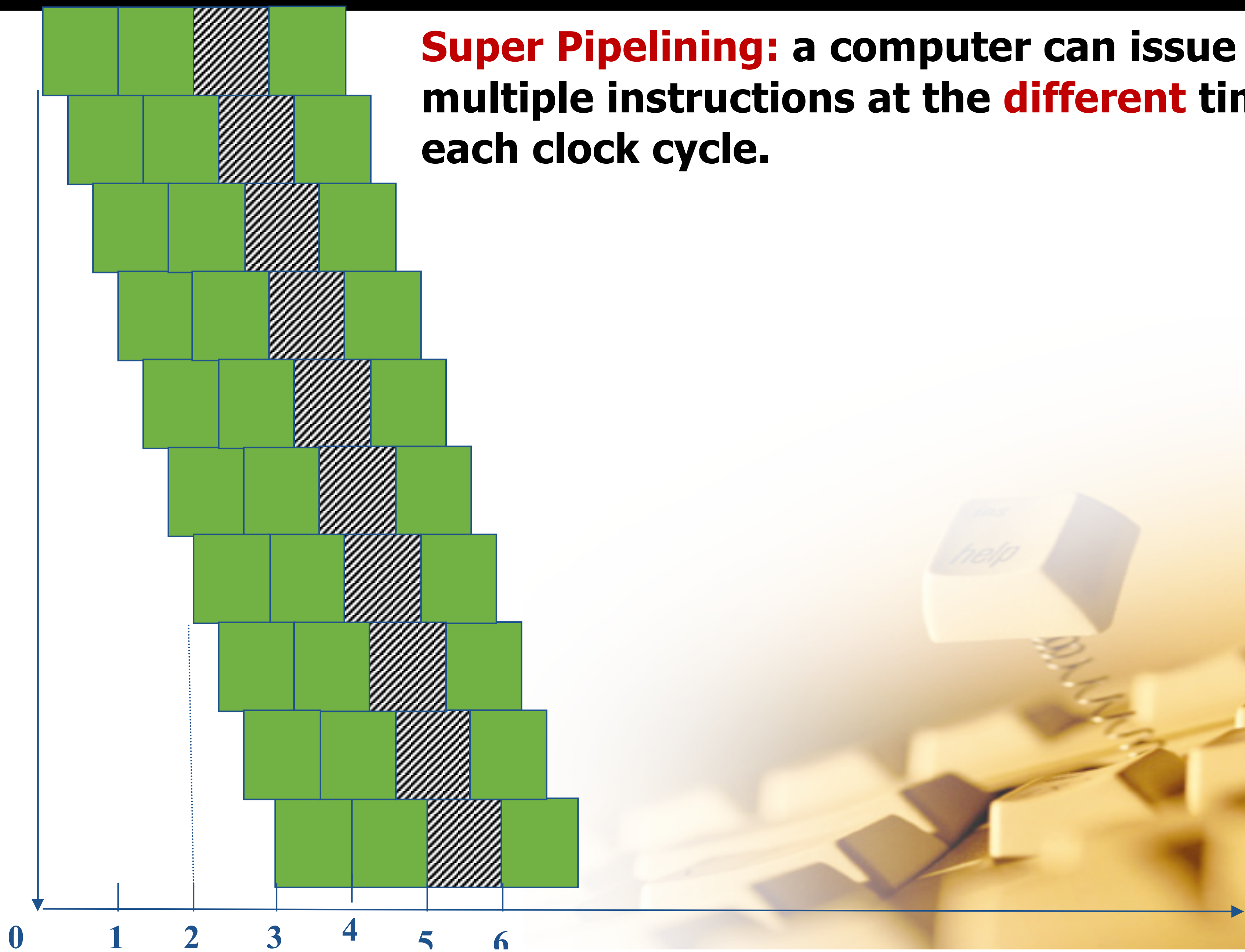
Superscalar: a computer can issue multiple instructions at the same time each clock cycle.

Chapter4 Pipelining Technology

VLIW: **a computer with very long instruction word**



Chapter4 Pipelining Technology



Super Pipelining: a computer can issue multiple instructions at the **different** time each clock cycle.

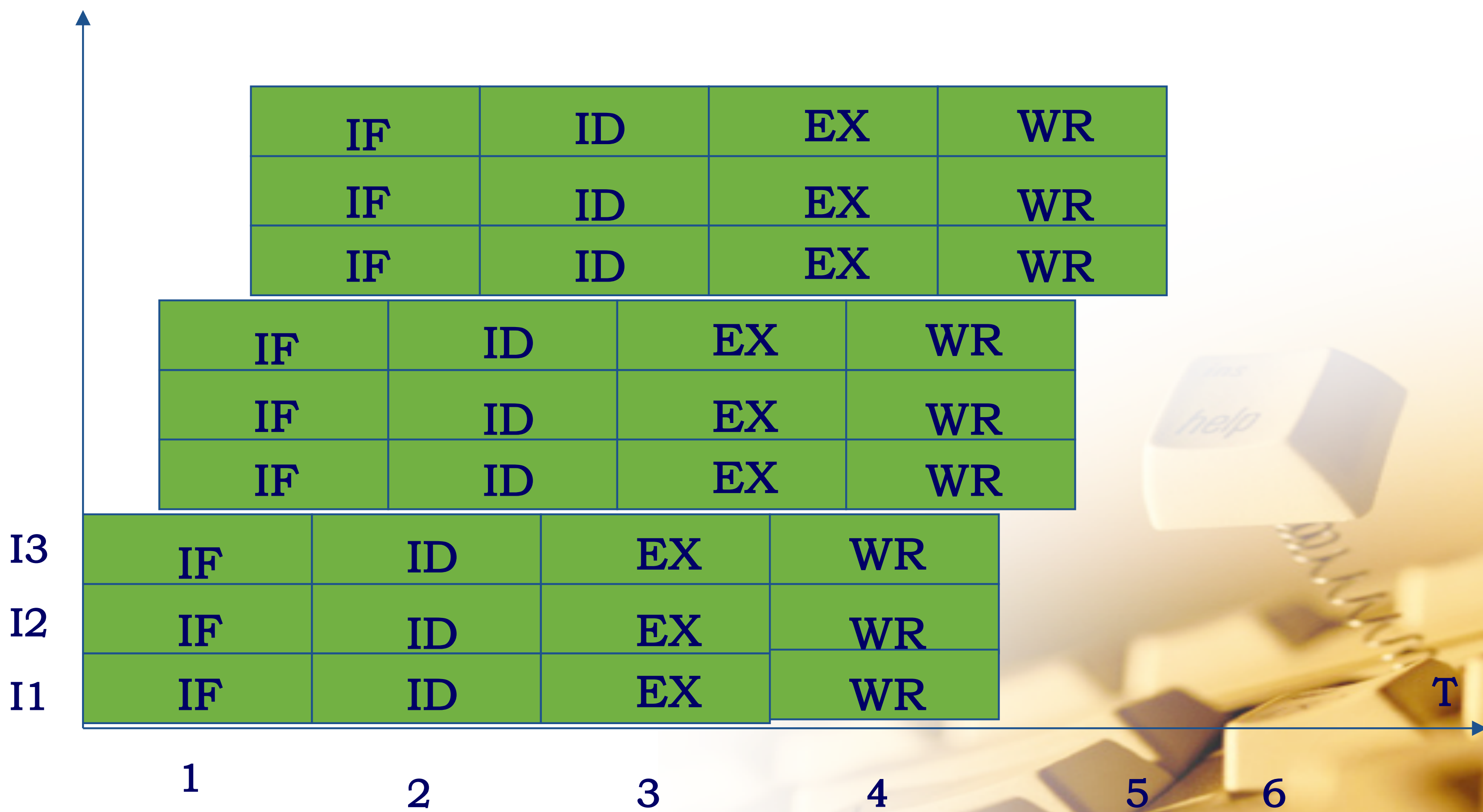
Example：MIPS R4000

IF	IS	RF	EX	DF	DS	TC	WB							
	IF	IS	RF	EX	DF	DS	TC	WB						
		IF	IS	RF	EX	DF	DS	TC	WB					
			IF	IS	RF	EX	DF	DS	TC	WB				
				IF	IS	RF	EX	DF	DS	TC	WB			
					IF	IS	RF	EX	DF	DS	TC	WB		
						IF	IS	RF	EX	DF	DS	TC	WB	
							IF	IS	RF	EX	DF	DS	TC	WB

Clock Cycle

Chapter4 Pipelining Technology

Superscalar & Super Pipelining



Vector Pipelining

◆ Vector Processing Method

Example: A*(B+C)

FORTRAN: Do 10 I=1,n
10 D（I）=A（I）*（B（I）+C（I））

1. Horizontal Processing Method

$d_1=a_1*(b_1+c_1)$

$d_2=a_2*(b_2+c_2)$

.

.

$d_n=a_n*(b_n+c_n)$

2. Vertical Processing Method

Bi+Ci→Ei

Ai*Ei→Di

3. Grouping Processing Method of Vertical and Horizontal

n=k*m+r

Bi+Ci→Ei (1~m)	⇒	Bi+Ci→Ei (m+1~2m)
Ei*Ai→Di (1~m)		Ei*Ai→Di (m+1~2m)

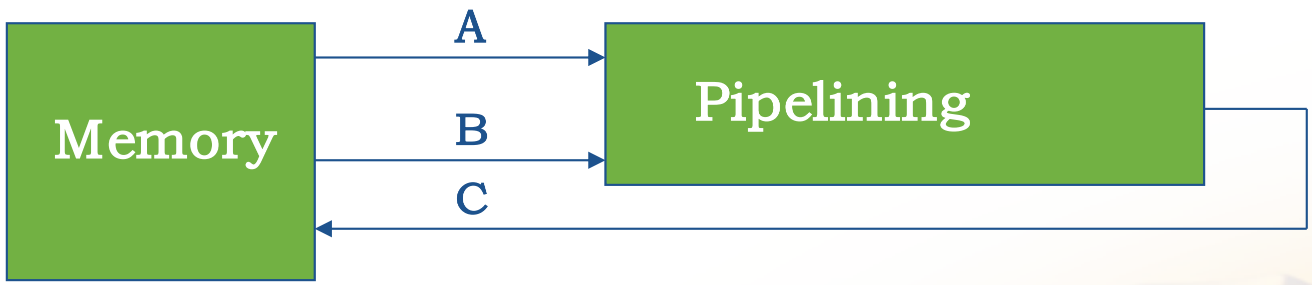
……

◆ Vector Processor Structure

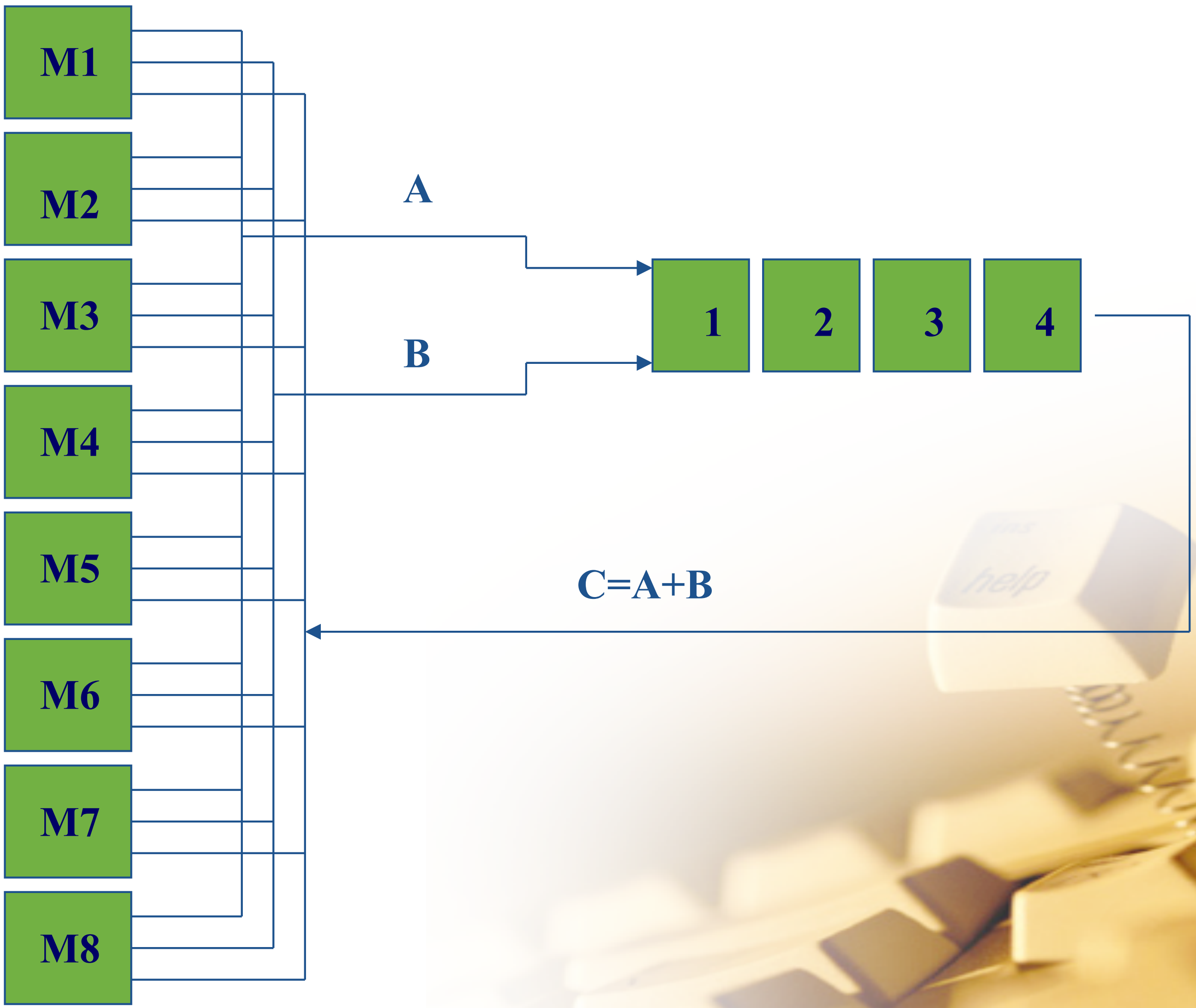
- **Memory-Memory**
- **Register-Register**

1. Memory-Memory Structure

Example: C=A+B



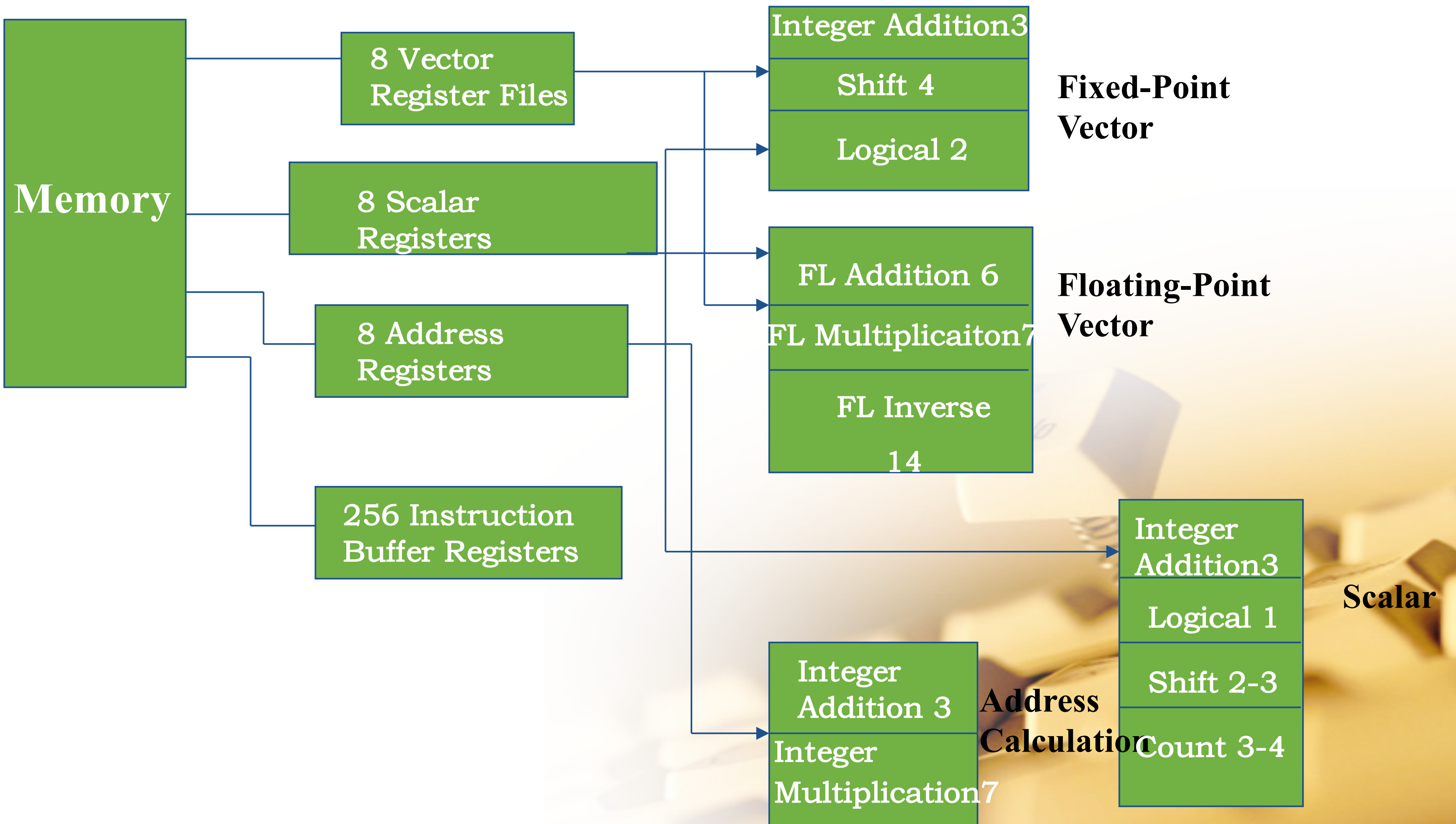
Chapter4 Pipelining Technology



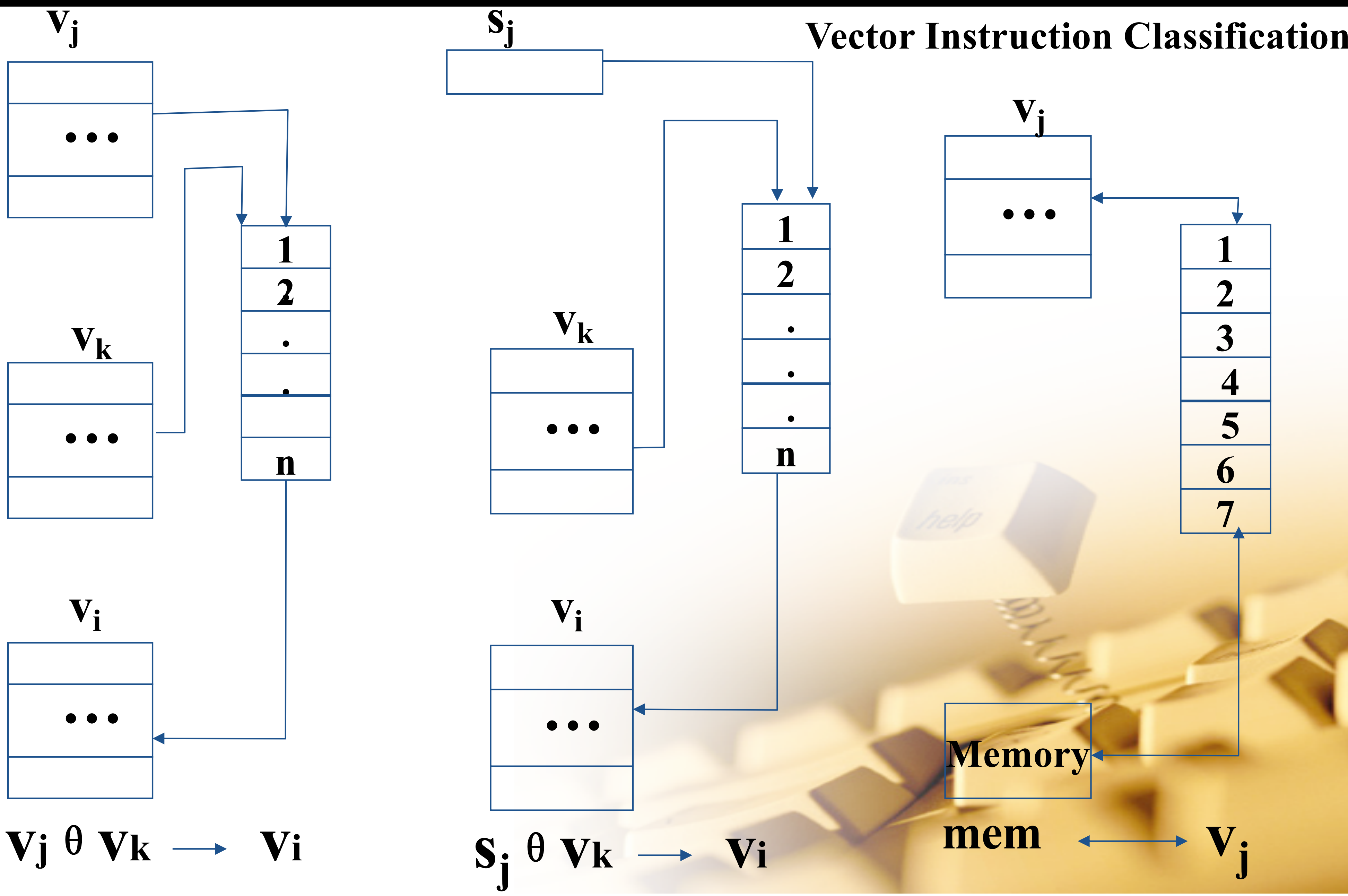
4						1	2	3	4	5	6	7	8		
3					1	2	3	4	5	6	7	8			
2				1	2	3	4	5	6	7	8				
1			1	2	3	4	5	6	7	8					
M8						RB6	RB6	RA8	RA8	WC4	WC4				
M7					RB5	RB5	RA7	RA7	WC3	WC3					
M6				RB4	RB4	RA6	RA6	WC2	WC2						
M5			RB3	RB3	RA5	RA5	WC1	WC1							
M4		RB2	RB2	RA4	RA4									WC8	WC8
M3	RB1	RB1	RA3	RA3									WC7	WC7	
M2		RA2	RA2					RB8	RB8			WC6	WC6		
M1	RA1	RA1					RB7	RB7			WC5	WC5			

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 t

2. Register-Register Structure (Cray-1)



Vector Instruction Classification



Chapter4 Pipelining Technology

◆ Vector Parallel Processing

❖ Full Parallel

V1+V2 →V3
V4*V5→V6

❖ Function Unit Conflict

V1+V2→V3
V4+V5→V6

❖ Vector Register （Vi） Conflict

V1+V2 →V3
V4*V2→V5

V1+V2→V3
V4*V5→V3

❖ Function Unit & Vi Conflict

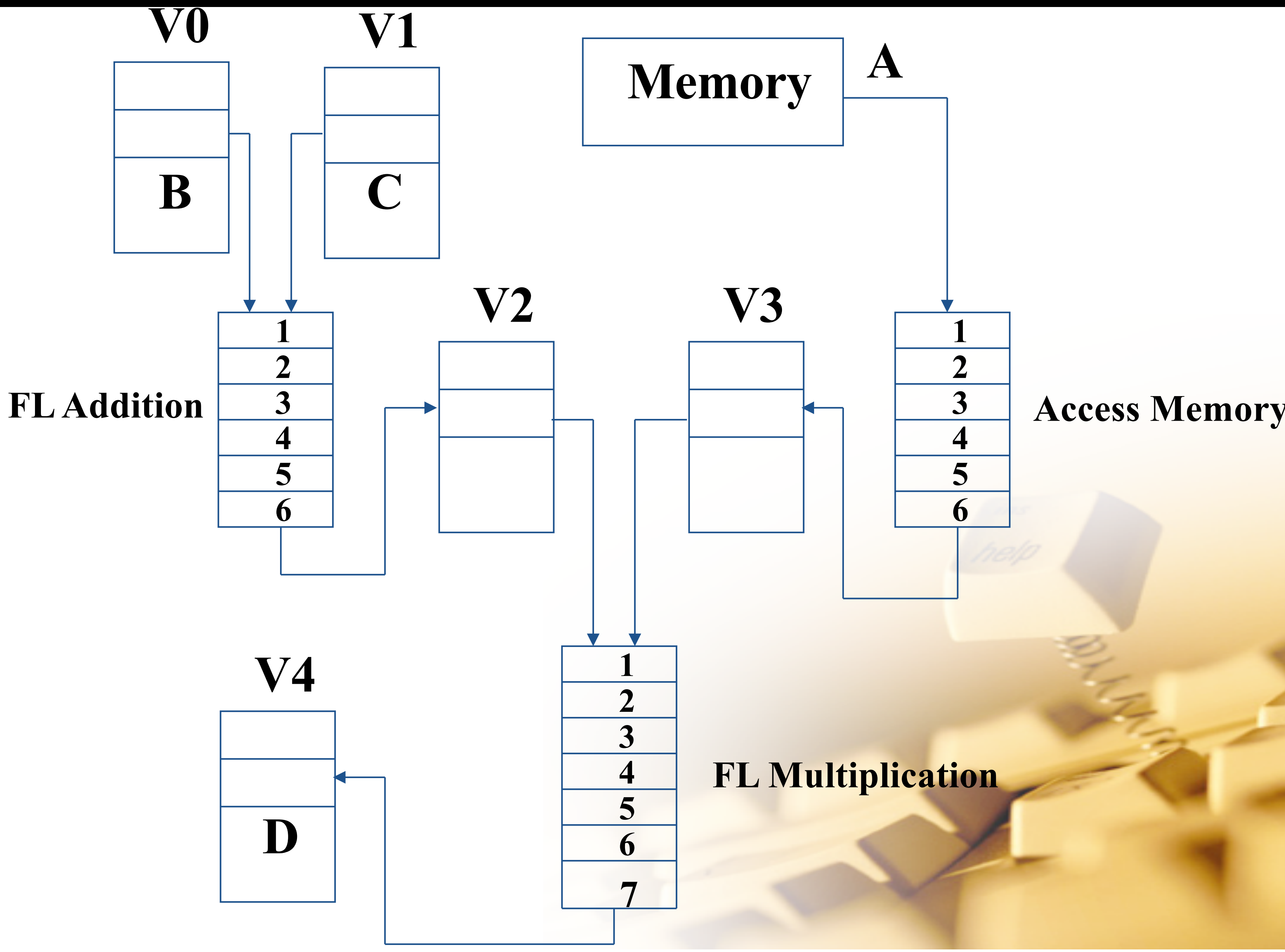
V1+V2 →V0
V1+V5 →V3

Vector Link Technology

1. RAW
2. Time Limitation

Example:

LD V3, A ; V3←A, Access Memory
ADDV V2, V0, V1 ; V2←V0+V1, Floating-Point Addition
MULTV V4, V2, V3 ; V4←V2*V3, Floating-Point Multiplication


V0
B
V1
C
Memory
A
V2
V3
FL Addition
1
2
3
4
5
6
Access Memory
1
2
3
4
5
6
V4
D
FL Multiplication
1
2
3
4
5
6
7

Assume that LD needs 6 $\triangle t$, ADDV needs 6 $\triangle t$ and MULTV needs 7 $\triangle t$. The set-up time of vector register and memory is both 1 $\triangle t$.

N- Vector Length

Full Serial：

$$[(1+6+1)+N-1]+[(1+6+1)+N-1]+[(1+7+1)+N-1]=(3N+22)\ \triangle t$$

The first and second instruction in parallel, serial with the third

$$[(1+6+1)+N-1]+[(1+7+1)+N-1]=(2N+15)\ \triangle t$$

The first and second instruction in parallel, link with the third

$$[(1+6+1)+(1+7+1)+(N-1)]=(N+16)\ \triangle t$$