

# 計算機論理設計

## 7. レジスタ回路設計

一色 剛

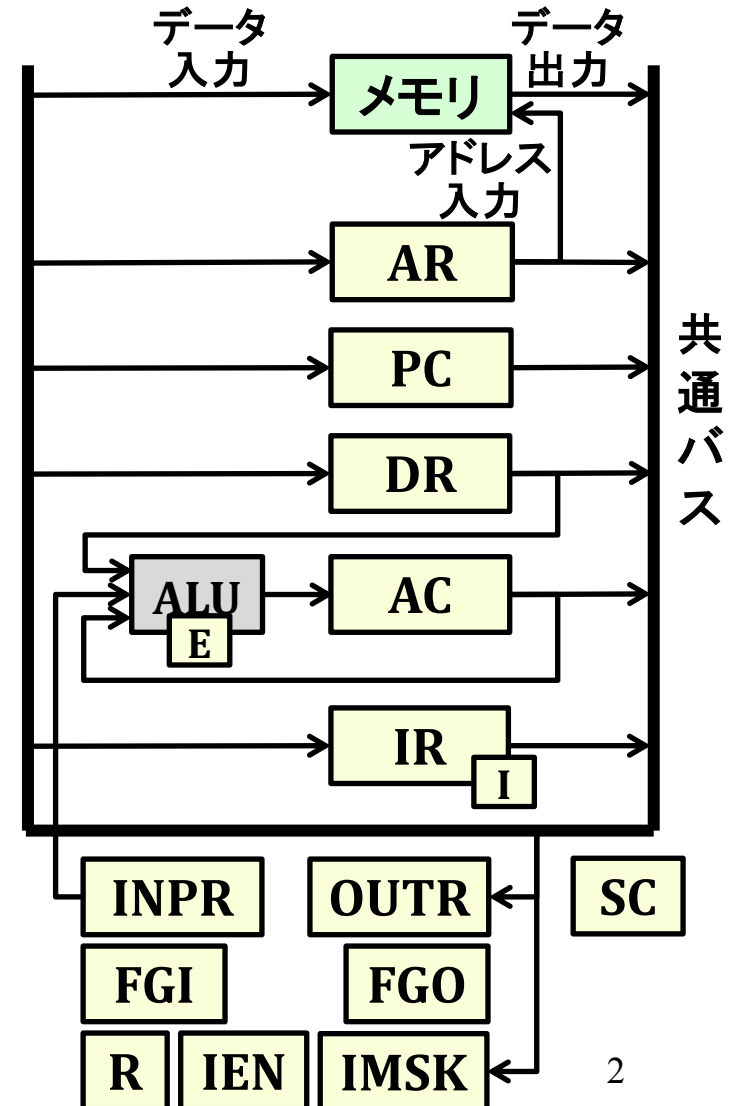
工学院情報通信系

[isshiki@ict.e.titech.ac.jp](mailto:isshiki@ict.e.titech.ac.jp)

# 計算機ハードウェアアーキテクチャ

## (入出力レジスタ・タイミング生成器含む)

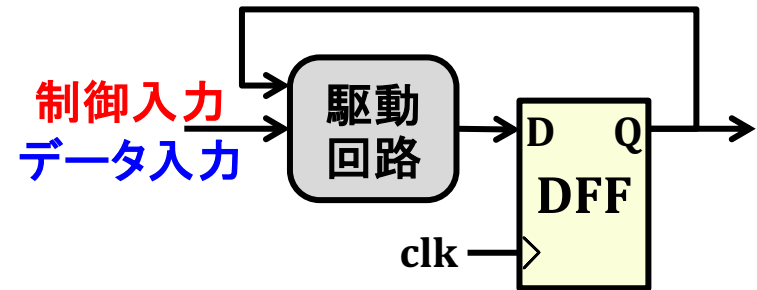
メモリ	アドレス: 12 bits、データ: 16 bits
AR (11:0)	アドレスレジスタ
PC (11:0)	プログラムカウンタ
DR (15:0)	データレジスタ
AC (15:0)	アキュムレータ
IR (15:0)	命令レジスタ
E	状態レジスタ
I	命令フラグレジスタ
SC (2:0)	タイミング生成器
INPR (7:0)	入力レジスタ
OUTR (7:0)	出力レジスタ
FGI	入力フラグレジスタ
FGO	出力フラグレジスタ
IEN	割込み制御レジスタ
IMSK(1:0)	割込みマスクレジスタ
R	割込み状態レジスタ



# レジスタ回路設計

## レジスタ機能

- 書込み(ロード) : (例)  $AR \leftarrow PC$
- クリア(0書込み) : (例)  $AR \leftarrow 0$
- インクリメント : (例)  $PC \leftarrow PC + 1$
- データ保持



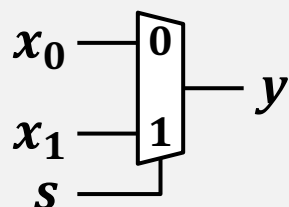
	ロード	クリア	インクリメント	保持
AR (11:0)	○	○	○	○
PC (11:0)	○	○	○	○
DR (15:0)	○		○	○
AC (15:0)	○	○	○	○
IR (15:0)	○			○
SC (2:0)		○	○	
INPR (7:0)	○			○
OUTR (7:0)	○			○
IMSK(1:0)	○			○

## レジスタ回路構成

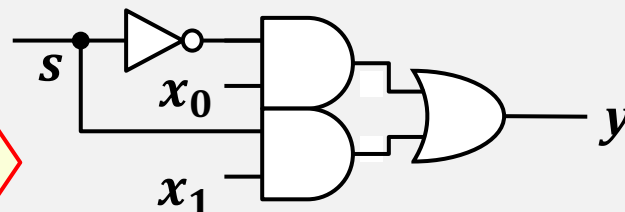
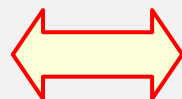
- Dフリップフロップ(DFF)
- 駆動回路(組合せ回路)

# Dフリップフロップ回路構成

## 2入力マルチプレクサ

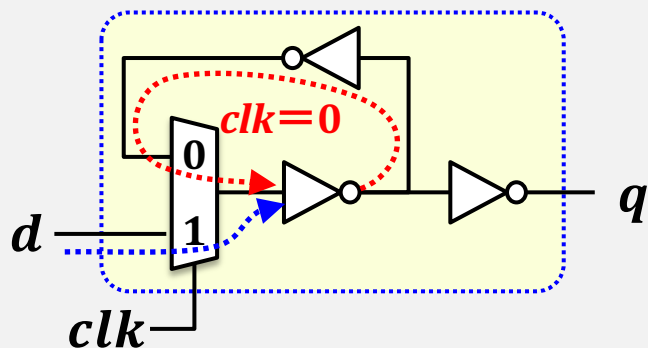


$$y = \begin{cases} x_0 & (s = 0) \\ x_1 & (s = 1) \end{cases}$$



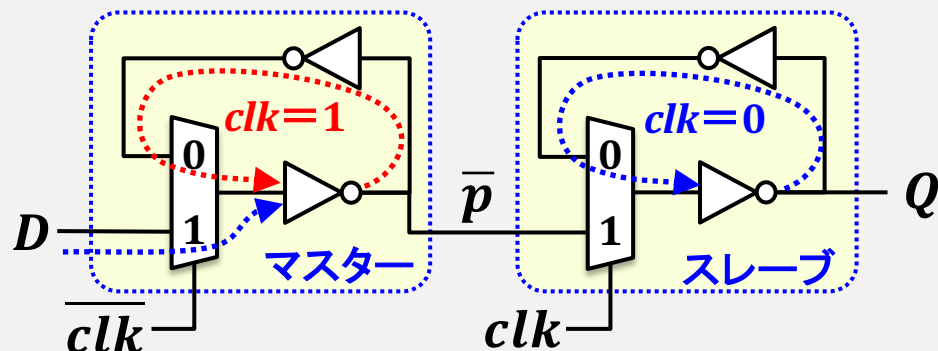
$$y = \bar{s} \cdot x_0 + s \cdot x_1$$

## Dラッチ (レベル駆動型)

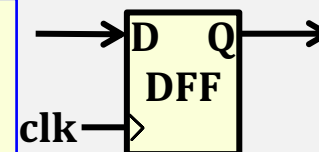


$$q = \begin{cases} d & (clk = 1) \\ \text{不変} & (clk = 0) \end{cases}$$

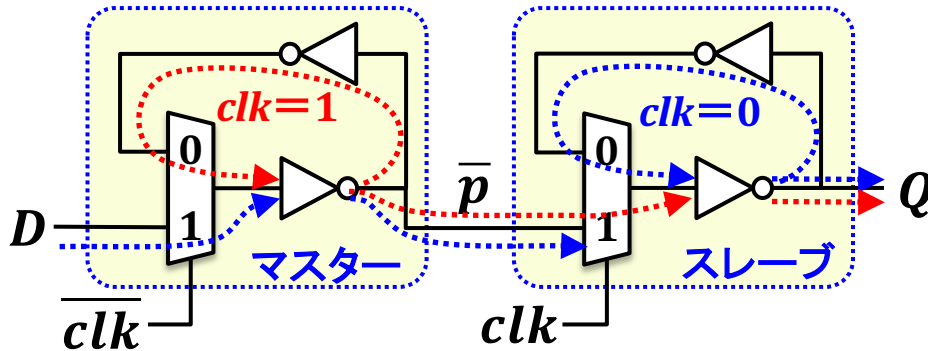
## Dフリップフロップ (エッジ駆動型)



$$Q = \begin{cases} D & (clk: 0 \rightarrow 1 \text{ 遷移}) \\ \text{不変} & (otherwise) \end{cases}$$



# Dフリップフロップ回路動作



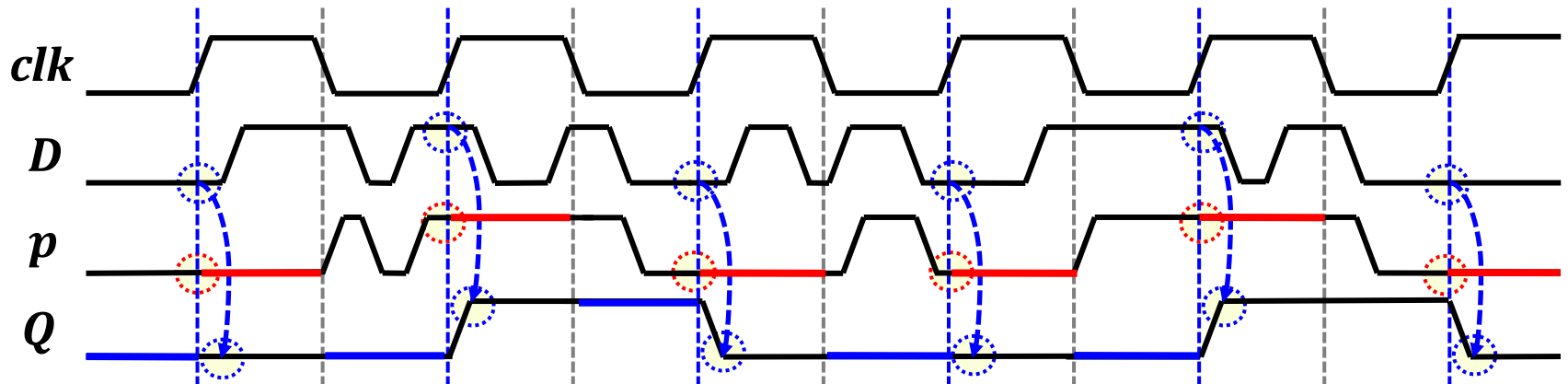
## Dフリップフロップ動作

- $clk$ 立上り時の $D$ 入力を $Q$ 出力に転送
- その他の区間では $Q$ は**不変**  
→  $clk$ 立上り時の **$D$ の値**を $Q$ に**保存**する(**メモリ機能**)

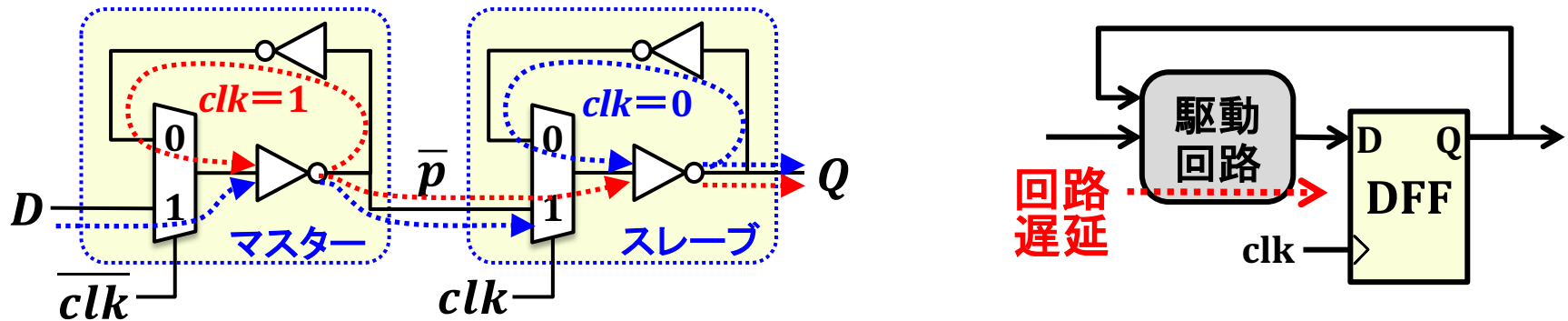
$$p = \begin{cases} D & (clk = 0) \\ \text{不変} & (clk = 1) \end{cases}$$

$$Q = \begin{cases} \text{不変} & (clk = 0) \\ p & (clk = 1) \end{cases}$$

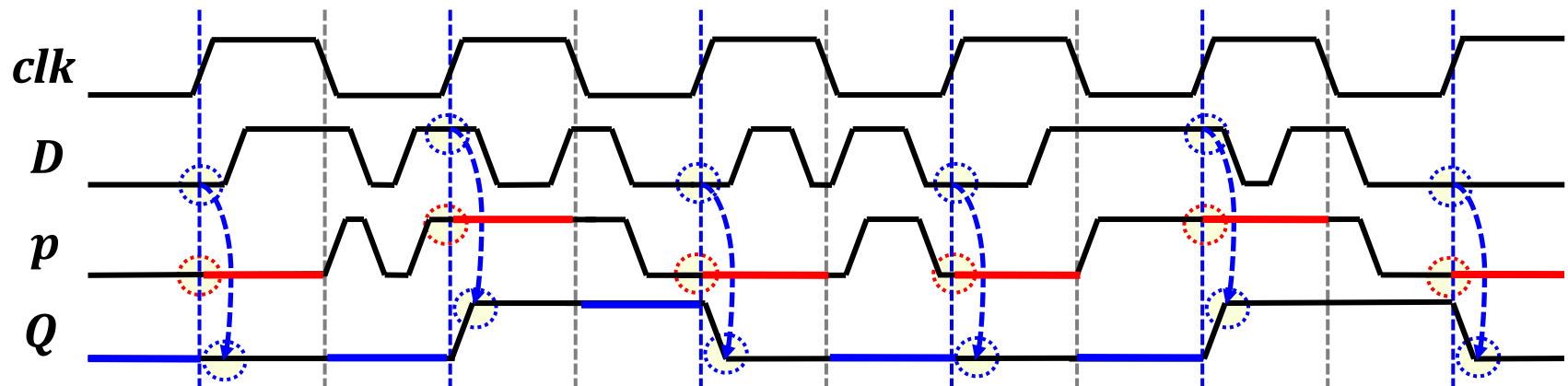
$$Q = \begin{cases} D & (clk: 0 \rightarrow 1 \text{ 遷移}) \\ \text{不変} & (\text{otherwise}) \end{cases}$$



# 順序回路の動作速度と回路遅延

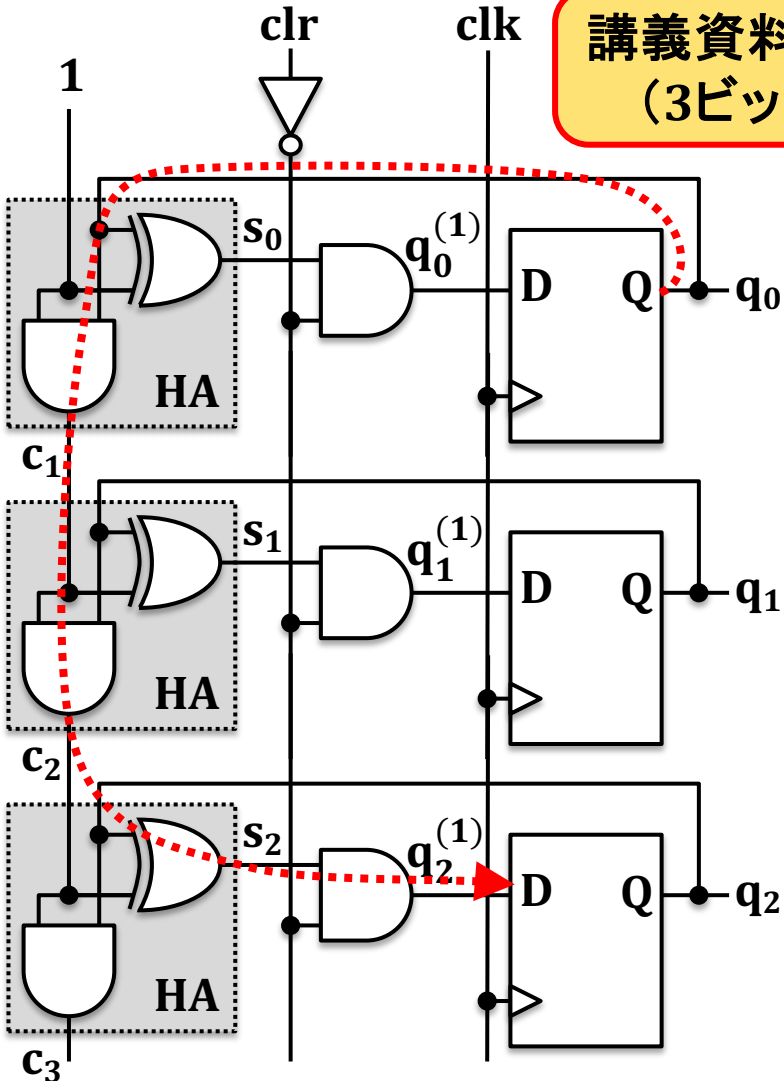


- 順序回路の動作速度：駆動回路（組合せ回路）の最大回路遅延で決まる
- 駆動回路出力（D入力）は、内部回路の回路遅延の差異によって、**1クロック周期で何度も変化することがある**（下図のようなD入力の变化は実際にあり得る）



# 駆動回路の最大回路遅延

## 講義資料5:P22 参照 (3ビットカウンタ)



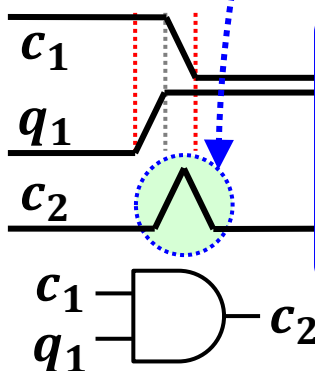
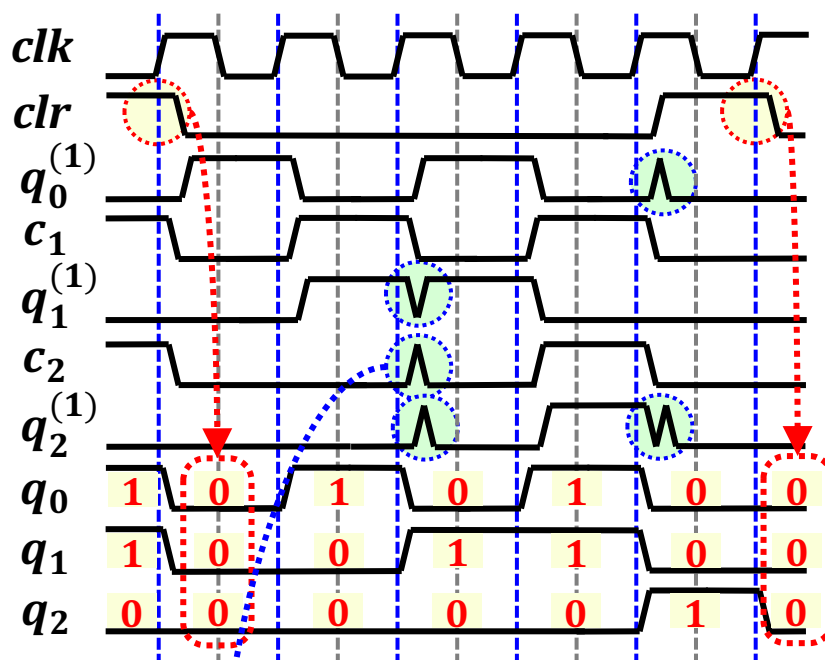
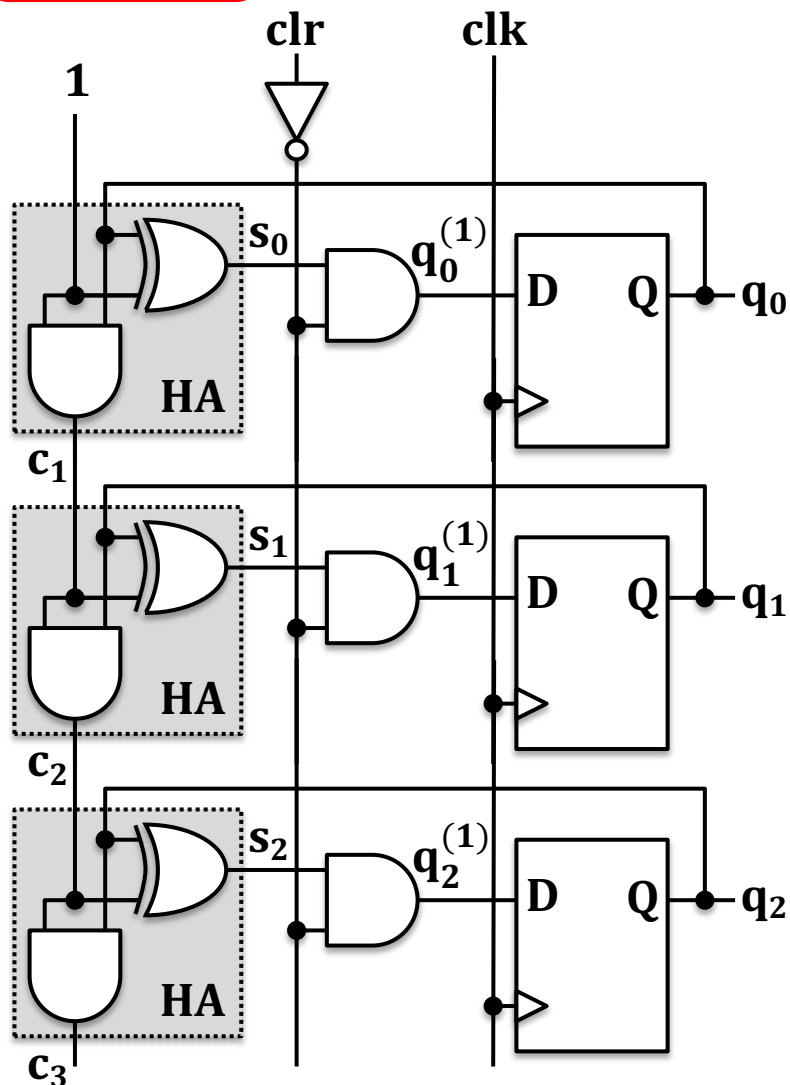
**順序回路の動作速度：駆動回路（組合せ回路）の最大回路遅延で決まる**

**駆動回路の最大回路遅延：任意のDFFのQ出力から任意のD入力までの信号伝搬経路の回路遅延の最大値**

→ 1クロック周期を最大回路遅延よりも大きくしないと順序回路が誤動作する

→ ビット幅が大きい加算器は、  
キャリー伝搬経路が最大回路  
遅延部分になりやすい

# 3ビットカウンタの動作

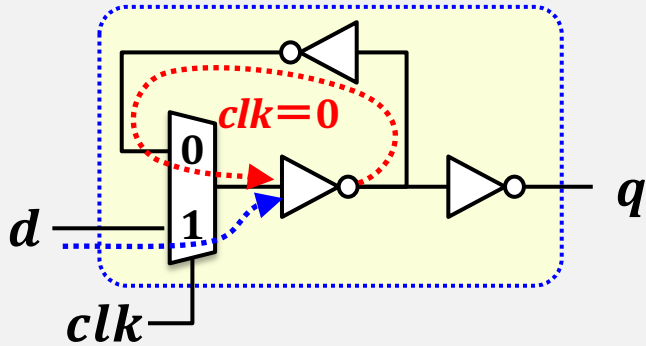


入力変化の時間的ずれ  
によって出力信号にス  
パイク(「ひげ」: **glitch**)  
が発生することがある



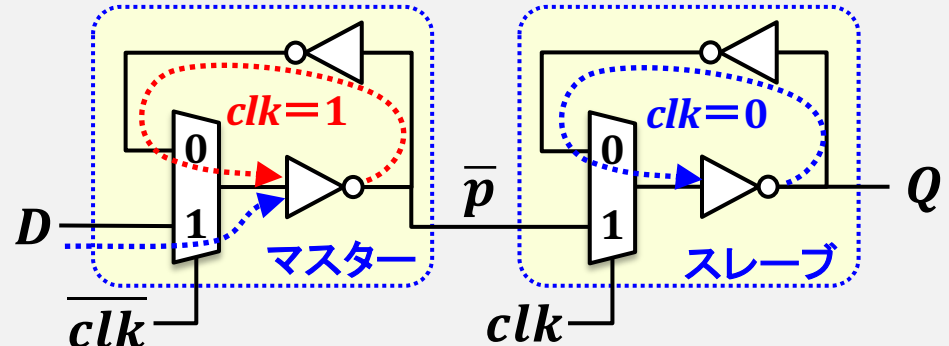
# ロード機能付きレジスタ

## Dラッチ (レベル駆動型)

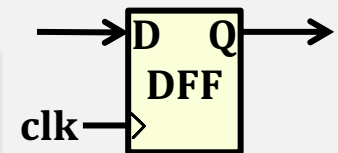


データをクロック半周期分  
保持 (clk=0の時)

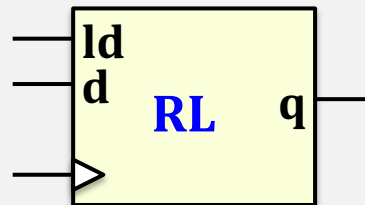
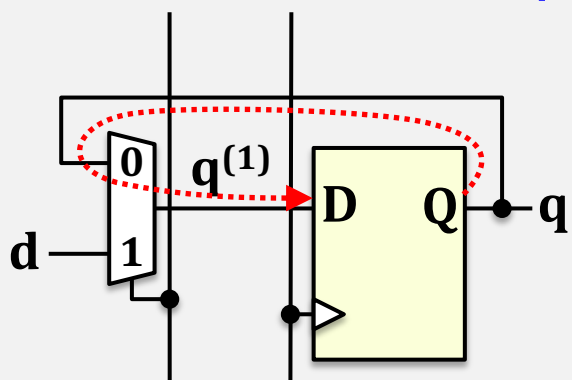
## Dフリップフロップ (エッジ駆動型)



データをクロック1周期分  
保持 (clk立上りに書き込み)



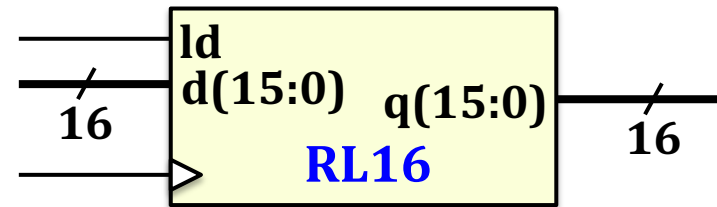
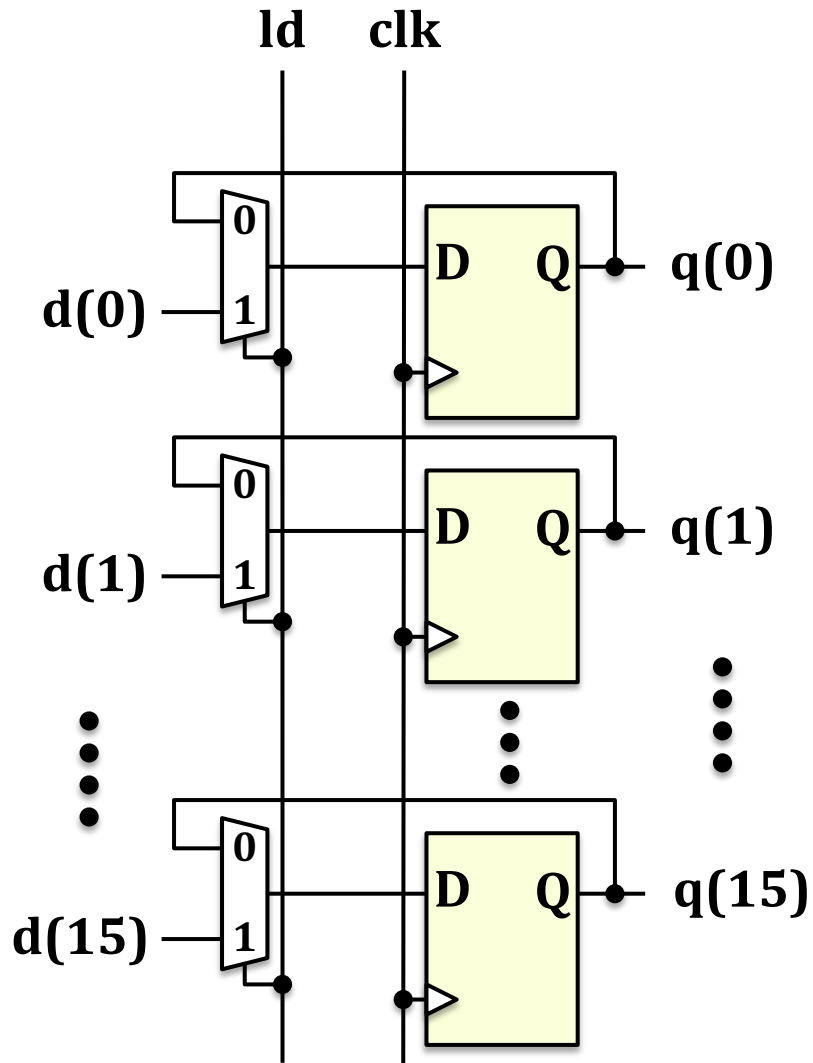
## ロード機能付きレジスタ



ld	$q^{(1)}$ (qの次状態)
1	d (書き込み)
0	q (不変)

データを任意期間 保存  
(ld = 1に書き込み)

# ロード機能付きレジスタ



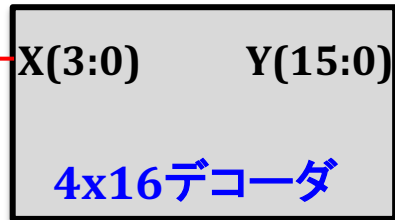
	ロード	保持
IR (15:0)	○	○
INPR (7:0)	○	○
OUTR (7:0)	○	○
IMSK(1:0)	○	○

# レジスタファイルの書込み制御

標準マイクロプロセッサでは多数の汎用レジスタ(レジスタファイル)を搭載

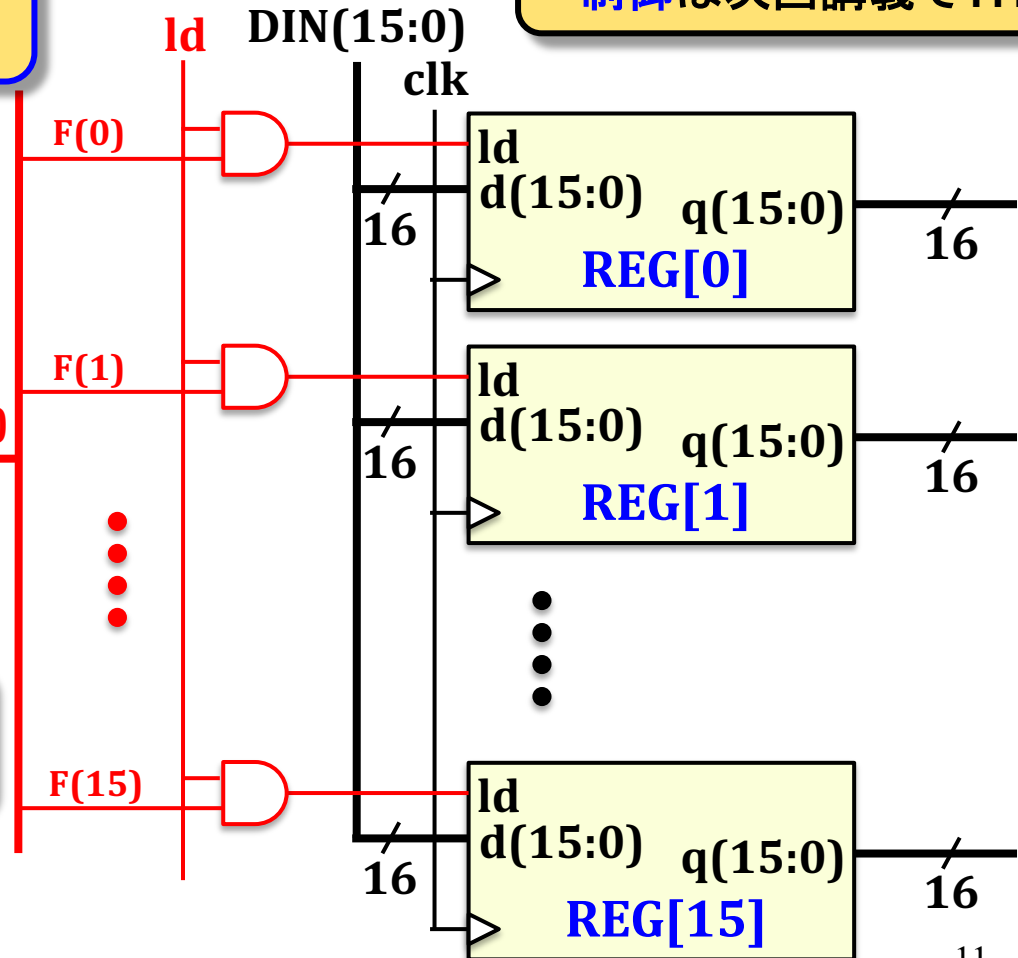
ロードするレジスタ番号を4ビット(**RID**)で指定

**RID(3:0)**



本講義の16ビットマイクロプロセッサにはレジスタファイルはない

レジスタファイルの**読出し**制御は次回講義で...



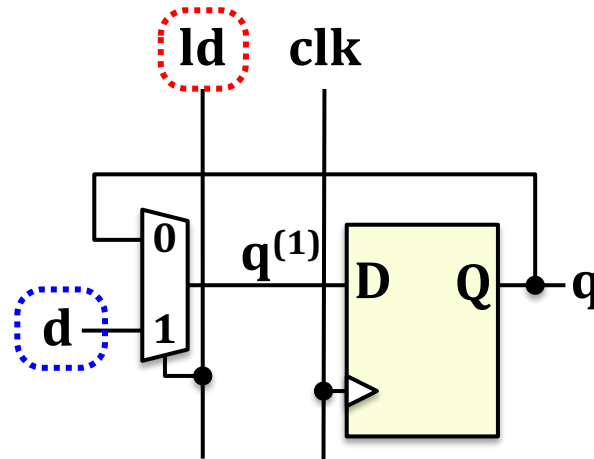
# 駆動回路の制御信号

ロード機能付きレジスタ

ld	$q^{(1)}$ (qの次状態)
1	d(書込み)
0	q(不変)

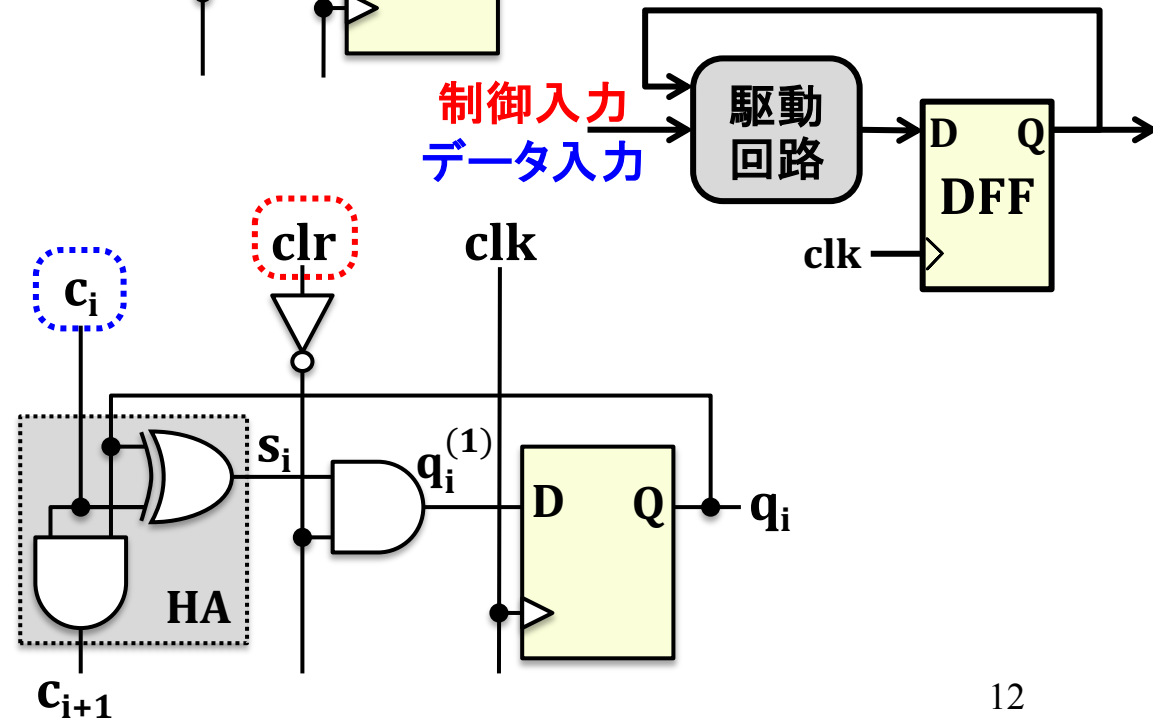
クリア機能付きカウンタ  
(タイミング生成回路)

clr	$q^{(1)}$ (qの次状態)
1	0
0	$\begin{cases} q + 1 \\ q_i^{(1)} = q_i \oplus c_i \\ c_{i+1} = q_i \cdot c_i \\ c_0 = 1 \end{cases}$

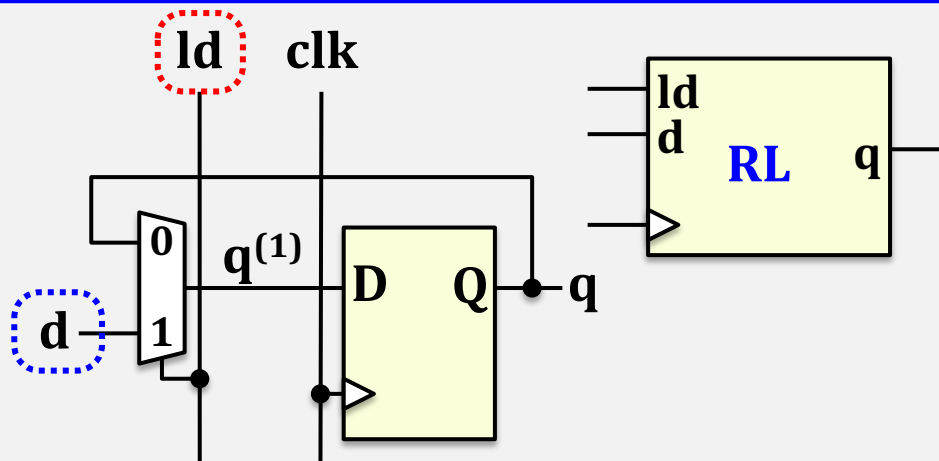


制御入力：全ビット  
で共通の入力信号

データ入力：各ビット  
で異なる入力信号

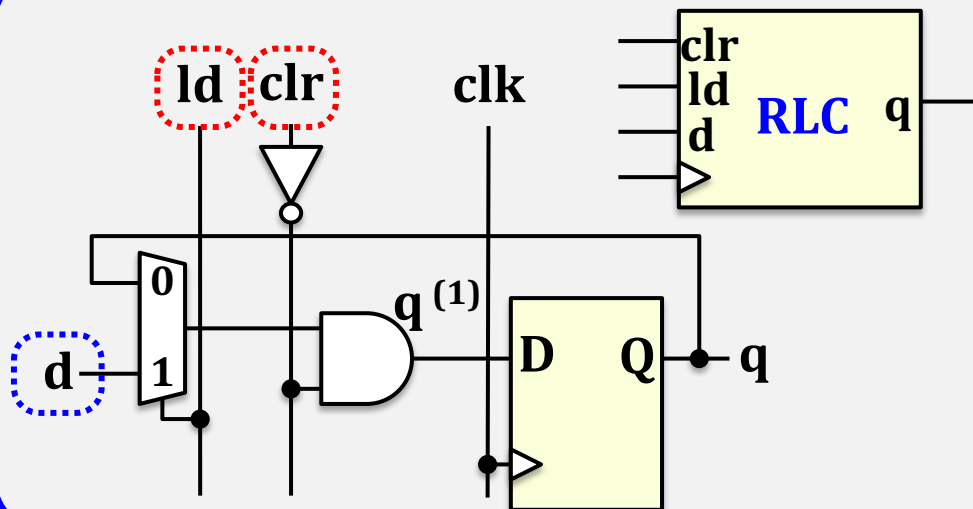


# ロード・クリア機能付きレジスタ



ロード機能付きレジスタ

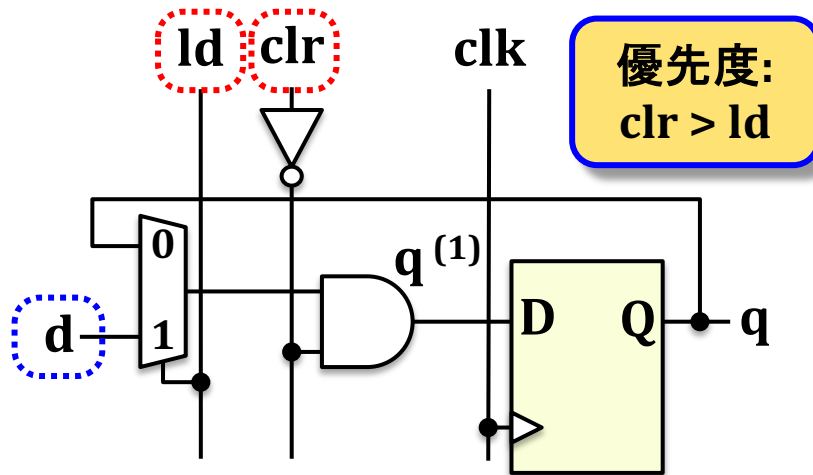
ld	$q^{(1)}$ (qの次状態)
1	d(書込み)
0	q(不変)



ロード・クリア機能付き  
レジスタ

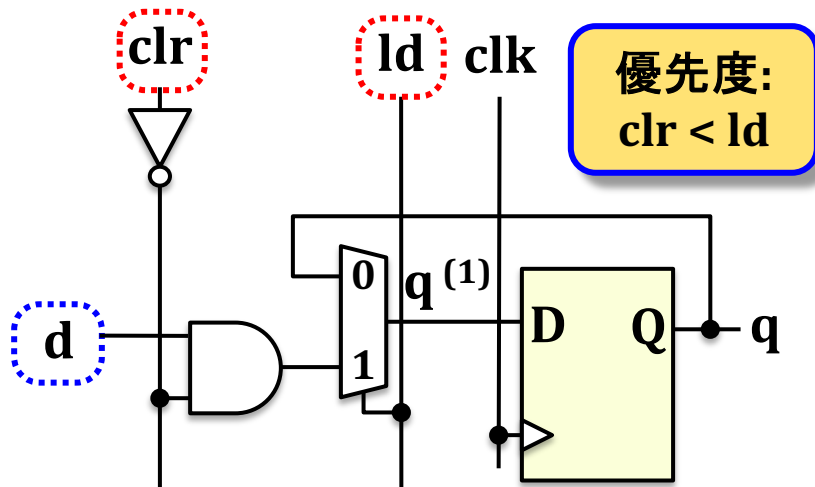
clr	ld	$q^{(1)}$ (qの次状態)
1	*	0
0	1	d(書込み)
0	0	q(不変)

# 制御信号の優先度と回路構成



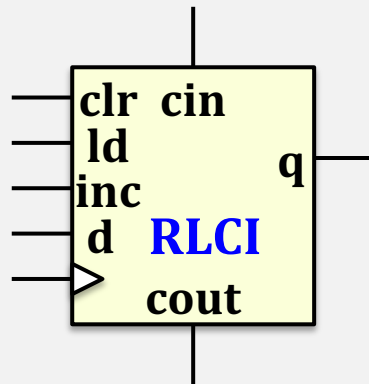
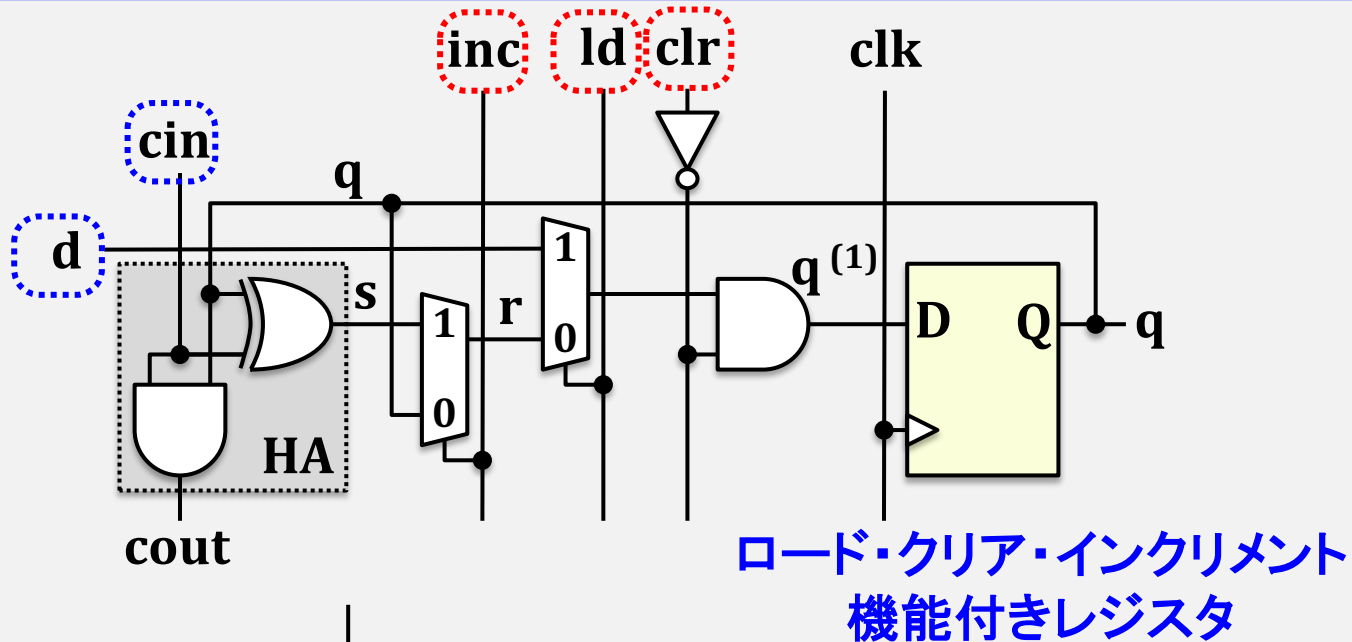
clr	ld	$q^{(1)}$ (qの次状態)
1	*	0
0	1	d(書込み)
0	0	q(不変)

**制御入力 が複数ある場合:**  
制御入力の優先度によって動作仕様と回路構成が変わる



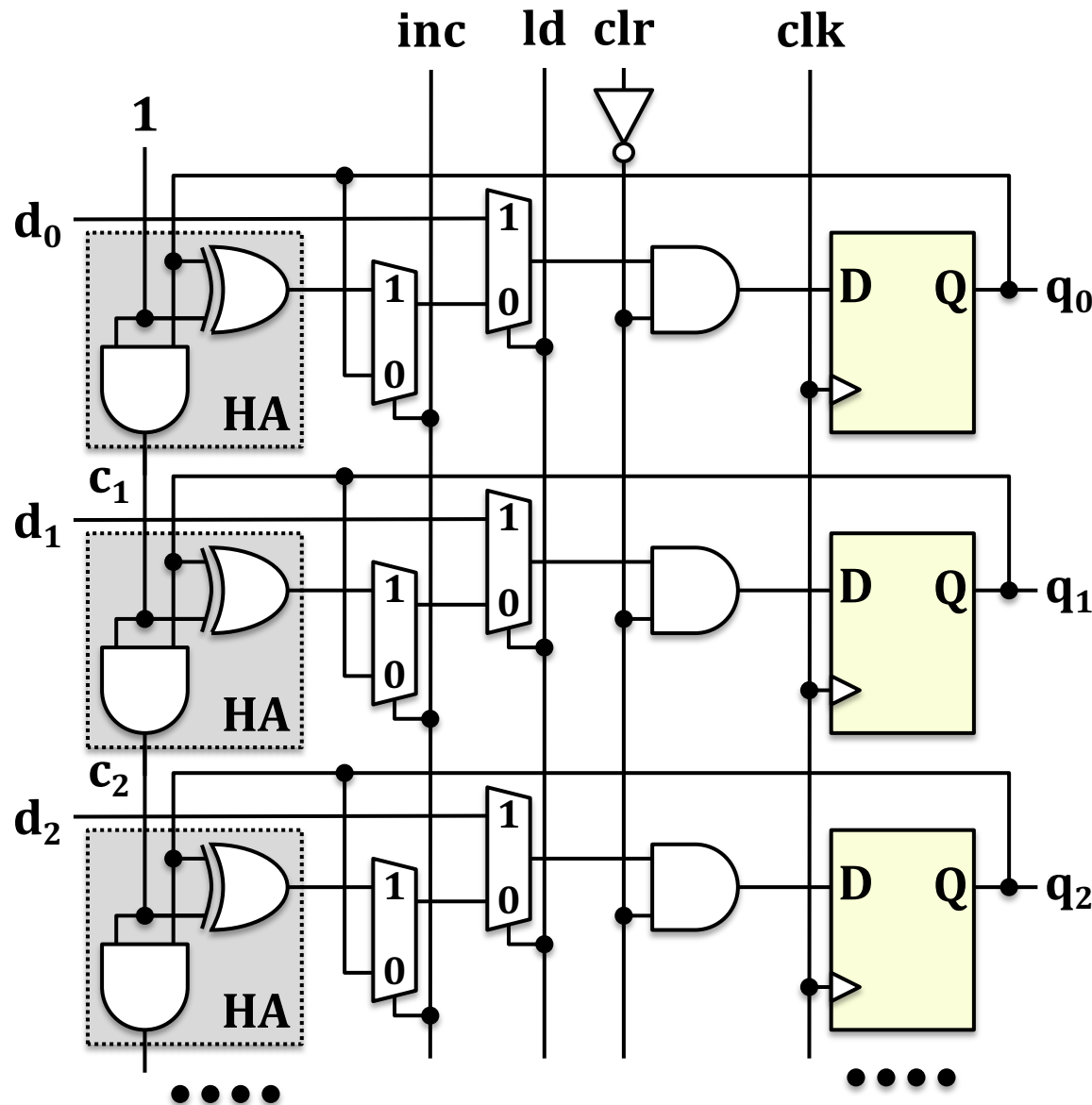
clr	ld	$q^{(1)}$ (qの次状態)
1	1	0
0	1	d(書込み)
*	0	q(不変)

# ロード・クリア・インクリメント機能付きレジスタ



clr	ld	inc	$q^{(1)}$ ( $q$ の次状態)
1	*	*	0
0	1	*	d(書込み)
0	0	1	$q+1$ (インクリメント)
0	0	0	$q$ (不変)

# ロード・クリア・インクリメント機能付きレジスタ



	ロード	クリア	インクリメント	保持
AR	○	○	○	○
PC	○	○	○	○
DR	○		○	○
AC	○	○	○	○



# 制御論理回路設計

## ■ 制御論理回路設計：

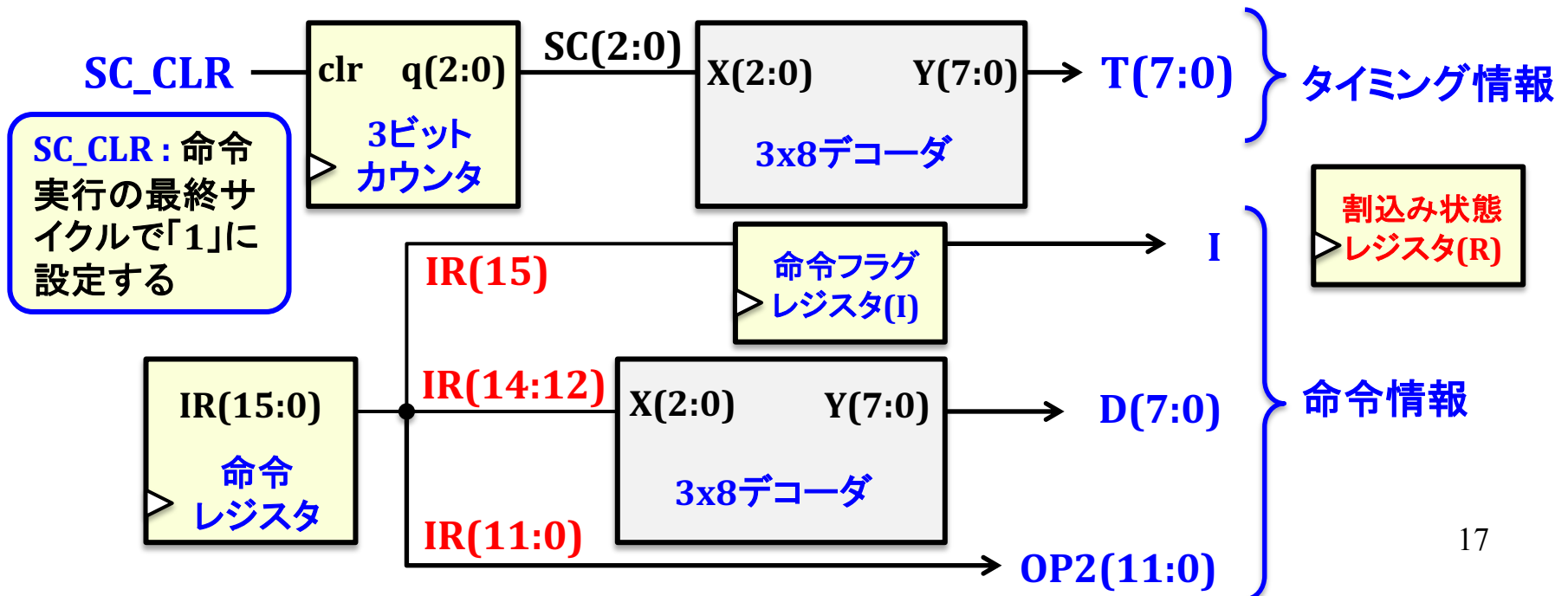
- レジスタ駆動回路制御入力の論理式を命令実行サイクル記述から導出

## ■ 制御論理回路の入力変数：

- $T(7:0)$ ：タイミング情報 → 3ビットカウンタSC(2:0)の3x8デコーダ出力
- $I$ ：命令フラグレジスタ
- $D(7:0)$ ：IR(14:12)の3x8デコーダ出力
- $OP2(11:0) = IR(11:0)$
- $R$ ：割込み状態レジスタ ←

講義資料5：P24参照

講義資料6：P10参照



# レジスタ転送記述による命令実行動作記述

レジスタ転送記述形式

講義資料5 : P26参照

**<条件式> : <レジスタ転送式>**

- 条件式 : <レジスタ転送式> を実行するための条件を表した論理式
- レジスタ転送式 : レジスタ・メモリへの代入式(コンマ「,」で複数の式を連結)

割込み検知

$$INTR = IEN \cdot (IMSK(0) \cdot FGI + IMSK(1) \cdot FGO)$$
$$INTR \cdot \overline{T(0)} \cdot \overline{T(1)} \cdot \overline{T(2)} : R \leftarrow 1$$

割込みサイクル

$$R \cdot T(0) : AR \leftarrow 0$$
$$R \cdot T(1) : M[AR] \leftarrow PC, PC \leftarrow 0$$
$$R \cdot T(2) : PC \leftarrow PC + 1, IEN \leftarrow 0, \\ R \leftarrow 0, SC \leftarrow 0$$

命令フェッチサイクル

$$\overline{R} \cdot T(0) : AR \leftarrow PC$$
$$\overline{R} \cdot T(1) : IR \leftarrow M[AR], PC \leftarrow PC + 1$$
$$\overline{R} \cdot T(2) : I \leftarrow IR(15), AR \leftarrow IR(11:0)$$

$SC \leftarrow 0$  : タイミング生成器初期化

# レジスタ転送記述による命令実行動作記述

レジスタ参照命令実行サイクル

	$r = D(7) \cdot \bar{I} \cdot T(3)$
CLA	$r \cdot OP2(11) : AC \leftarrow 0$
CLE	$r \cdot OP2(10) : E \leftarrow 0$
CMA	$r \cdot OP2(9) : AC \leftarrow \overline{AC}$
CME	$r \cdot OP2(8) : E \leftarrow \overline{E}$
CIR	$r \cdot OP2(7) : AC(14:0) \leftarrow AC(15:1),$ $AC(15) \leftarrow E, E \leftarrow AC(0)$
CIL	$r \cdot OP2(6) : AC(15:1) \leftarrow AC(14:0),$ $AC(0) \leftarrow E, E \leftarrow AC(15)$
INC	$r \cdot OP2(5) : AC \leftarrow AC + 1$
SPA	$r \cdot OP2(4) : IF (AC(15) = 0) THEN$ $PC \leftarrow PC + 1$
SNA	$r \cdot OP2(3) : IF (AC(15) = 1) THEN$ $PC \leftarrow PC + 1$
SZA	$r \cdot OP2(2) : IF (AC = 0) THEN$ $PC \leftarrow PC + 1$
SZE	$r \cdot OP2(1) : IF (E = 0) THEN$ $PC \leftarrow PC + 1$
HLT	$r \cdot OP2(0) : HALT$

入出力命令実行サイクル

	$p = D(7) \cdot I \cdot T(3)$
INP	$p \cdot OP2(11) : AC(7:0) \leftarrow INPR, FGI \leftarrow 0$
OUT	$p \cdot OP2(10) : OUTR \leftarrow AC(7:0), FGO \leftarrow 0$
SKI	$p \cdot OP2(9) : IF (FGI = 1) THEN$ $PC \leftarrow PC + 1$
SKO	$p \cdot OP2(8) : IF (FGO = 1) THEN$ $PC \leftarrow PC + 1$
ION	$p \cdot OP2(7) : IEN \leftarrow 1$
IOF	$p \cdot OP2(6) : IEN \leftarrow 0$
IMK	$p \cdot OP2(5) : IMSK \leftarrow AC(1:0)$

非メモリ参照命令共通

$D(7) \cdot T(3) : SC \leftarrow 0$

$SC \leftarrow 0$  : タイミング生成器初期化

# レジスタ転送記述による命令実行動作記述

## メモリ参照命令実行サイクル

	$\overline{D(7)} \cdot I \cdot T(3) :$	$AR \leftarrow M[AR]$ (間接アドレスサイクル)
AND	$D(0) \cdot T(4) :$	$DR \leftarrow M[AR]$
	$D(0) \cdot T(5) :$	$AC \leftarrow AC \& DR, SC \leftarrow 0$
ADD	$D(1) \cdot T(4) :$	$DR \leftarrow M[AR]$
	$D(1) \cdot T(5) :$	$AC \leftarrow AC + DR, E \leftarrow C_{out}(16), SC \leftarrow 0$
LDA	$D(2) \cdot T(4) :$	$DR \leftarrow M[AR]$
	$D(2) \cdot T(5) :$	$AC \leftarrow DR, SC \leftarrow 0$
STA	$D(3) \cdot T(4) :$	$M[AR] \leftarrow AC, SC \leftarrow 0$
BUN	$D(4) \cdot T(4) :$	$PC \leftarrow AR, SC \leftarrow 0$
BSA	$D(5) \cdot T(4) :$	$M[AR] \leftarrow PC, AR \leftarrow AR + 1$
	$D(5) \cdot T(5) :$	$PC \leftarrow AR, SC \leftarrow 0$
ISZ	$D(6) \cdot T(4) :$	$DR \leftarrow M[AR]$
	$D(6) \cdot T(5) :$	$DR \leftarrow DR + 1$
	$D(6) \cdot T(6) :$	$M[AR] \leftarrow DR, IF(DR = 0) THEN PC \leftarrow PC + 1, SC \leftarrow 0$

$SC \leftarrow 0$  : タイミング生成器初期化

# タイミング生成器クリア制御



**SC\_CLR**: 命令実行の最終  
サイクルで「1」に設定する

種別	条件	動作
割込みサイクル	$R \cdot T(2)$	$SC \leftarrow 0$
AND	$D(0) \cdot T(5)$	$SC \leftarrow 0$
ADD	$D(1) \cdot T(5)$	$SC \leftarrow 0$
LDA	$D(2) \cdot T(5)$	$SC \leftarrow 0$
STA	$D(3) \cdot T(4)$	$SC \leftarrow 0$
BUN	$D(4) \cdot T(4)$	$SC \leftarrow 0$
BSA	$D(5) \cdot T(5)$	$SC \leftarrow 0$
ISZ	$D(6) \cdot T(6)$	$SC \leftarrow 0$
非メモリ参照命令	$D(7) \cdot T(3)$	$SC \leftarrow 0$

$$SC\_CLR = R \cdot T(2) + D(7) \cdot T(3) + (D(3) + D(4)) \cdot T(4) + (D(0) + D(1) + D(2) + D(5)) \cdot T(5) + D(6) \cdot T(6)$$

# アドレスレジスタ(AR)制御

種別	条件	動作	制御信号
割込みサイクル	$R \cdot T(0) :$	$AR \leftarrow 0$	AR_CLR
命令フェッチサイクル	$\overline{R} \cdot T(0) :$	$AR \leftarrow PC$	AR_LD
	$\overline{R} \cdot T(2) :$	$AR \leftarrow IR(11:0)$	AR_LD
間接アドレスサイクル	$\overline{D(7)} \cdot I \cdot T(3) :$	$AR \leftarrow M[AR]$	AR_LD
BSA	$D(5) \cdot T(4) :$	$AR \leftarrow AR + 1$	AR_INC

$$AR\_CLR = R \cdot T(0)$$

$$AR\_LD = \overline{R} \cdot (T(0) + T(2)) + \overline{D(7)} \cdot I \cdot T(3)$$

$$AR\_INC = D(5) \cdot T(4)$$

# プログラムカウンタ(PC)制御

種別	条件	動作	制御信号
割込みサイクル	$R \cdot T(1) :$	$PC \leftarrow 0$	PC_CLR
BUN	$D(4) \cdot T(4) :$	$PC \leftarrow AR$	PC_LD
BSA	$D(5) \cdot T(5) :$	$PC \leftarrow AR$	PC_LD

$$PC\_CLR = R \cdot T(1)$$

$$PC\_LD = D(4) \cdot T(4) + D(5) \cdot T(5)$$

PCの初期値(0x10)の設定方法については次回以降の講義で説明する

# プログラムカウンタ(PC)制御

種別	条件	動作	制御信号
割込み	$R \cdot T(2) :$	$PC \leftarrow PC + 1$	PC_INC
命令フェッチ	$\bar{R} \cdot T(1) :$	$PC \leftarrow PC + 1$	PC_INC
ISZ	$D(6) \cdot T(6) :$	$IF (DR = 0) THEN PC \leftarrow PC + 1$	PC_INC
SPA	$r \cdot OP2(4) :$	$IF (AC(15) = 0) THEN PC \leftarrow PC + 1$	PC_INC
SNA	$r \cdot OP2(3) :$	$IF (AC(15) = 1) THEN PC \leftarrow PC + 1$	PC_INC
SZA	$r \cdot OP2(2) :$	$IF (AC = 0) THEN PC \leftarrow PC + 1$	PC_INC
SZE	$r \cdot OP2(1) :$	$IF (E = 0) THEN PC \leftarrow PC + 1$	PC_INC
SKI	$p \cdot OP2(9) :$	$IF (FGI = 1) THEN PC \leftarrow PC + 1$	PC_INC
SKO	$p \cdot OP2(8) :$	$IF (FGO = 1) THEN PC \leftarrow PC + 1$	PC_INC

$$r = D(7) \cdot \bar{I} \cdot T(3)$$

$$p = D(7) \cdot I \cdot T(3)$$

$$(AC = 0) \leftrightarrow \overline{AC(0)} \cdot \overline{AC(1)} \cdots \overline{AC(15)}$$

$$\begin{aligned}
 PC\_INC = & R \cdot T(2) + \bar{R} \cdot T(1) + D(6) \cdot T(6) \cdot (DR = 0) \\
 & + r \cdot \left( OP2(4) \cdot \overline{AC(15)} + OP2(3) \cdot AC(15) + OP2(2) \cdot (AC = 0) + OP2(1) \cdot \bar{E} \right) \\
 & + p \cdot (OP2(9) \cdot FGI + OP2(8) \cdot FGO)
 \end{aligned}$$



# その他レジスタ制御

種別	条件	動作	制御信号
命令フェッチサイクル	$\bar{R} \cdot T(1) :$	$IR \leftarrow M[AR]$	IR_LD
AND	$D(0) \cdot T(4) :$	$DR \leftarrow M[AR]$	DR_LD
ADD	$D(1) \cdot T(4) :$	$DR \leftarrow M[AR]$	DR_LD
LDA	$D(2) \cdot T(4) :$	$DR \leftarrow M[AR]$	DR_LD
ISZ	$D(6) \cdot T(4) :$	$DR \leftarrow M[AR]$	DR_LD
	$D(6) \cdot T(5) :$	$DR \leftarrow DR + 1$	DR_INC
OUT	$p \cdot OP2(10) :$	$OUTR \leftarrow AC(7:0)$	OUTR_LD
IMK	$p \cdot OP2(5) :$	$IMSK \leftarrow AC(1:0)$	IMSK_LD

$$IR\_LD = \bar{R} \cdot T(1)$$

$$OUTR\_LD = p \cdot OP2(10)$$

$$IMSK\_LD = p \cdot OP2(5)$$

$$DR\_LD = (D(0) + D(1) + D(2) + D(6)) \cdot T(4)$$

$$DR\_INC = D(6) \cdot T(5)$$

$$DR\_CLR = 0$$

その他のレジスタ制御等については次回以降の講義で説明する