

本テキストや授業のビデオなどの電子ファイルを他人に転送したり、ネットへアップロードすることなどを禁止します。



論理設計

東野担当11回目

授業スライド

12月16日4限

基礎工学部情報科学科 東野輝夫





授業計画

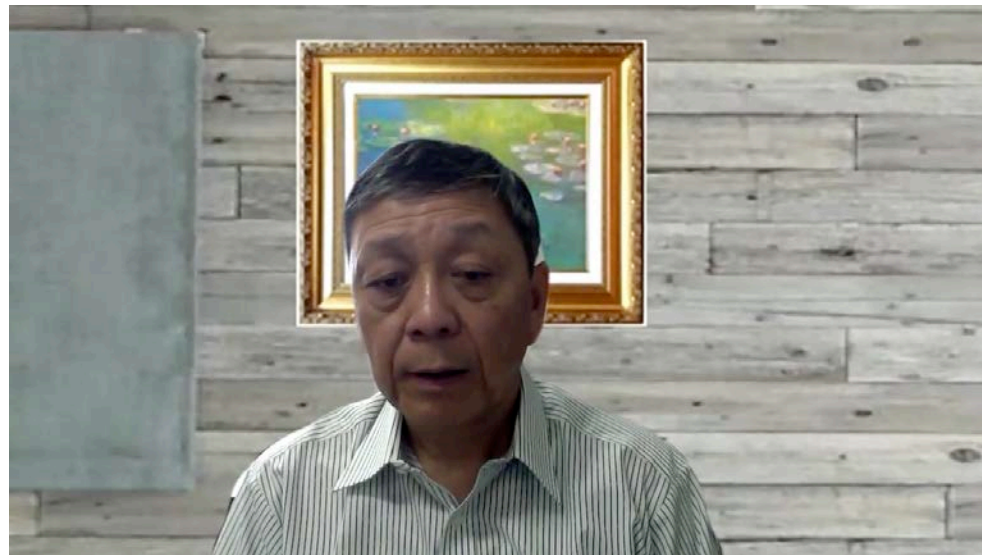
- 授業計画：

1. ドントケアを含む論理関数の簡単化（6章）
2. フリップフロップとレジスタ（10章）
3. 同期式順序回路（Mealy型, Moore型順序回路）（11章）
4. カルノー図を用いた論理関数の簡単化（1章から5章の復習）
5. 組合せ論理回路設計、よく用いられる組み合わせ回路（7章, 8章）
6. 加減算器とALU、順序回路の簡単化（9章, 12章前半）
7. 演習
8. 順序回路の簡単化、カウンタ（12章後半, 13章）
9. 中間試験（1章～11章）
10. ICを用いた順序回路の実現（15章）
11. 演習
12. CPUの設計（付録）
13. CPUの設計, 演習
14. 乗算器と除算器（14章）
15. 期末試験（12章～15章, 付録）



質問について

- メールで随時問い合わせや質問にお答えしますので、何かあれば、higashino@ist.osaka-u.ac.jp までメールで質問して下さい。
- また、時間を決めてZoomなどを用いて質問にお答えすることも可能ですので、まずはメールで疑問点や問い合わせ事項などを連絡して下さい。





お願い

本テキストや授業のビデオなどの電子ファイルを他人に転送したり、ネットへアップロードすることなどを禁止します。

著作権保護

- この授業のテキスト（教科書）や授業スライド、授業ビデオの著作権保護に努めて下さい。
- この授業のビデオやスナップショットを録画したり，それらを他の人に転送したり，インターネット上で公開したりすることを禁止します。
- この授業で利用するスライドにはオーム社の教科書の図などが含まれているので，著作権保護の観点から，この授業スライドの公開につながる行為は謹んでください。
- 来年度は CLE を使ったメディア授業でなく，対面の授業ができることを期待していますが，今年度の演習課題の解答が事前に公開されたりすると，来年度の授業で同じ演習課題が使えなくなり，授業テキストの大幅な修正が必要になるため，協力をお願いします。



第15章 ICを用いた順序回路の 実現 (演習)






第15章

ICを用いた順序回路の実現

- 演習問題

演習 問題

- 1 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ,

- 2 与えられた数 n の平方根 \sqrt{n} を求める同期式順序回路を設計せよ.





二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ
 - 2進数の掛け算 ($1101 * 1011 = 10001111 = 13 * 11 = 143$)

$$\begin{array}{r}
 \times \quad 13 \\
 \hline
 11 \\
 \hline
 13 \\
 13 \\
 \hline
 143
 \end{array}$$

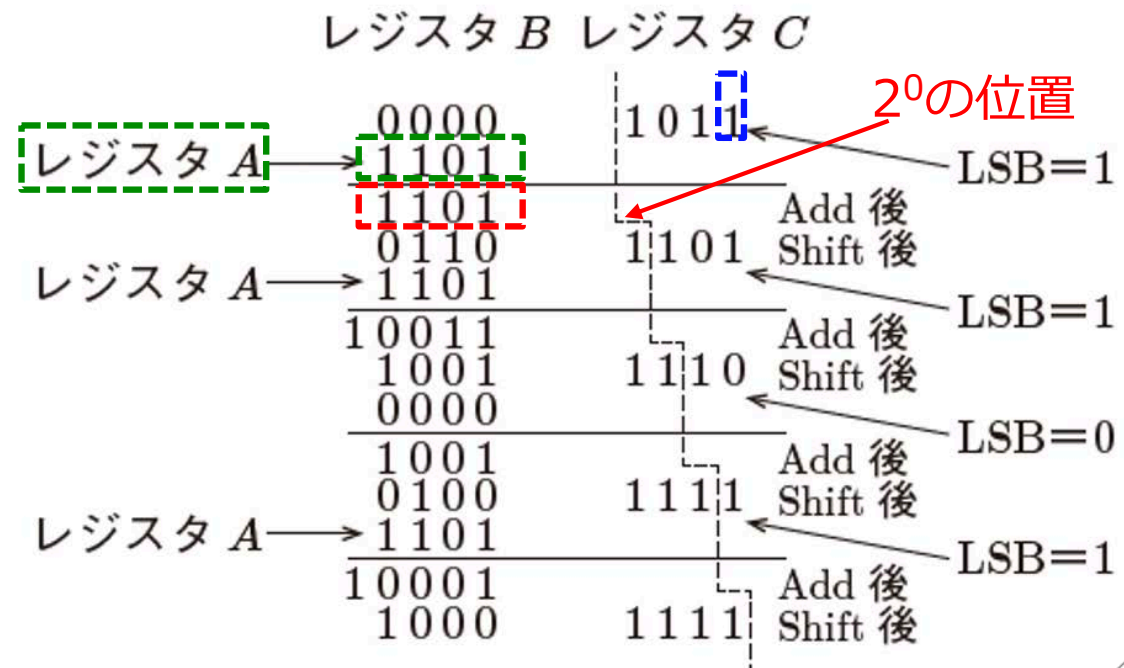
$$\begin{array}{r}
 \times \quad 1101 \\
 \hline
 1011 \\
 1101 \\
 0000 \\
 1101 \\
 \hline
 10001111
 \end{array}$$



二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ
 - 2進数の掛け算 ($1101 * 1011 = 10001111 = 13 * 11 = 143$)

$$\begin{array}{r}
 \times \quad \begin{array}{c} \boxed{1101} \\ 1011 \\ \boxed{1101} \end{array} \\
 \hline
 1101 \\
 0000 \\
 1101 \\
 \hline
 10001111
 \end{array}$$

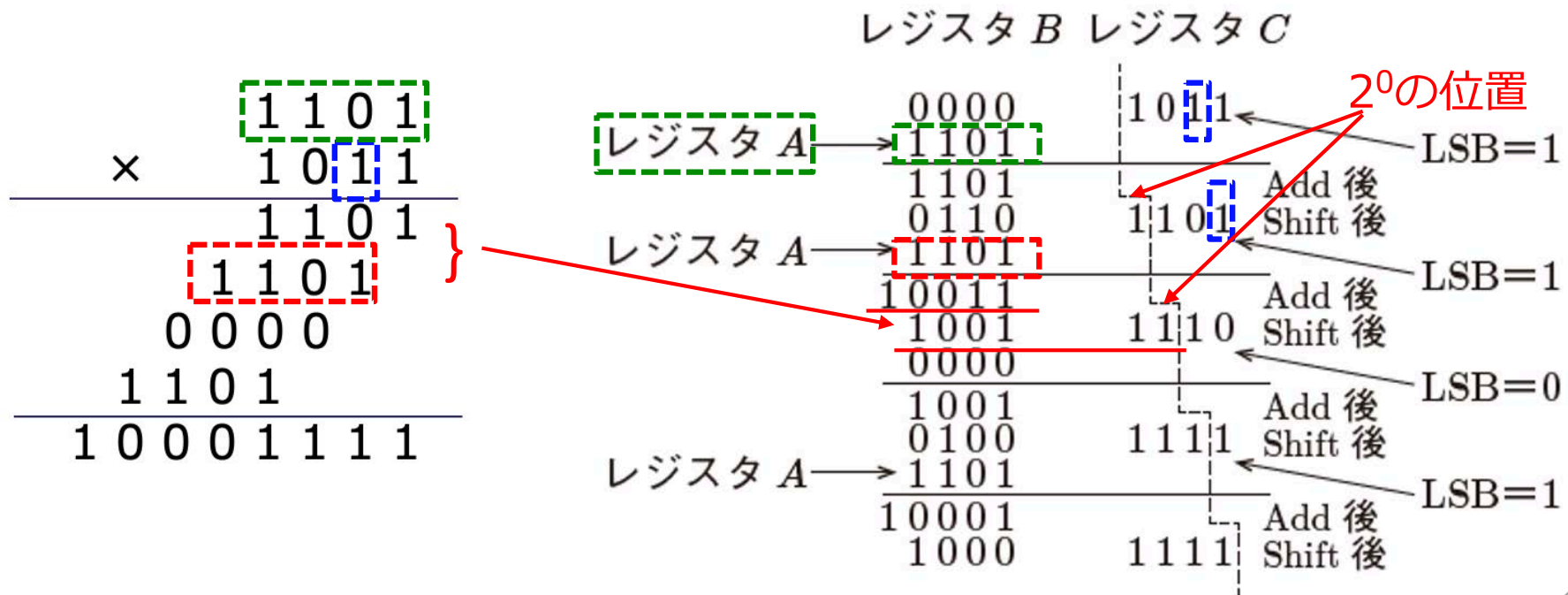


解図 15・2

$N=1101$, $M=1011$ の積 $K=10001111$ を計算するアルゴリズムの概略

二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ
 - 2進数の掛け算 ($1101 * 1011 = 10001111 = 13 * 11 = 143$)

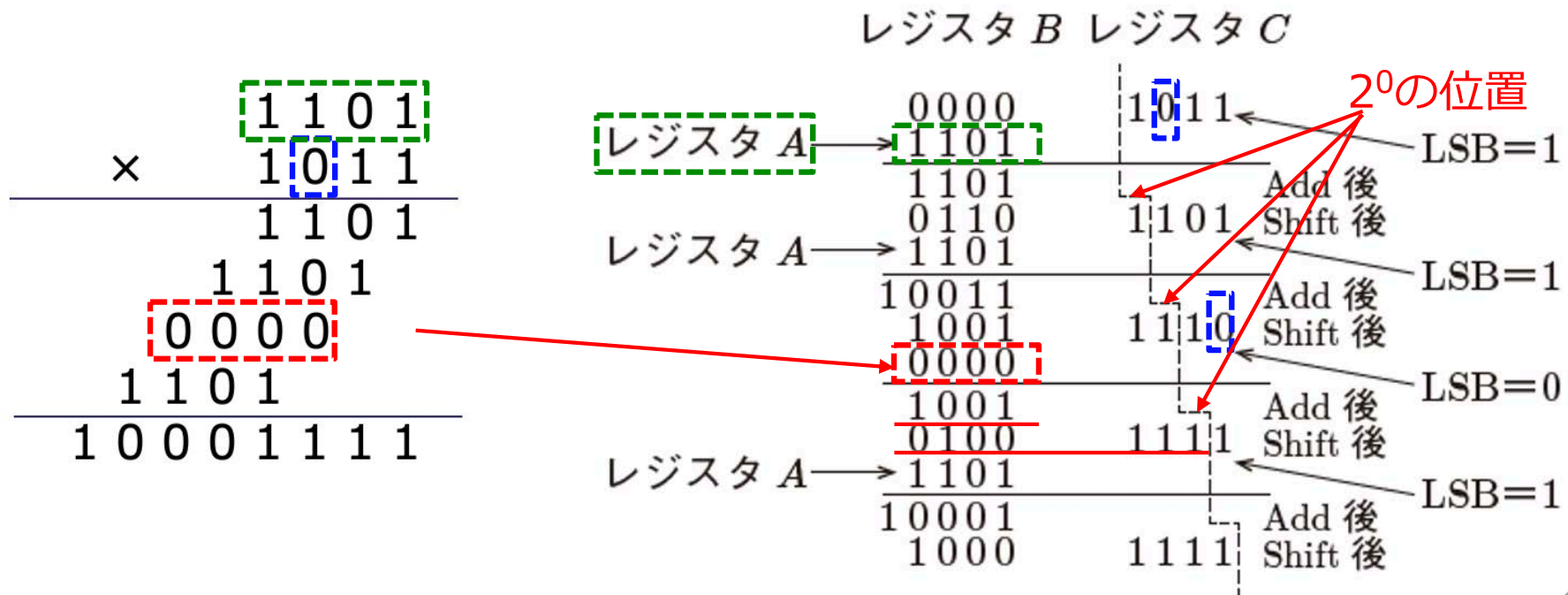


解図 15・2

$N=1101$, $M=1011$ の積 $K=10001111$ を計算するアルゴリズムの概略

二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ
 - 2進数の掛け算 ($1101 * 1011 = 10001111 = 13 * 11 = 143$)

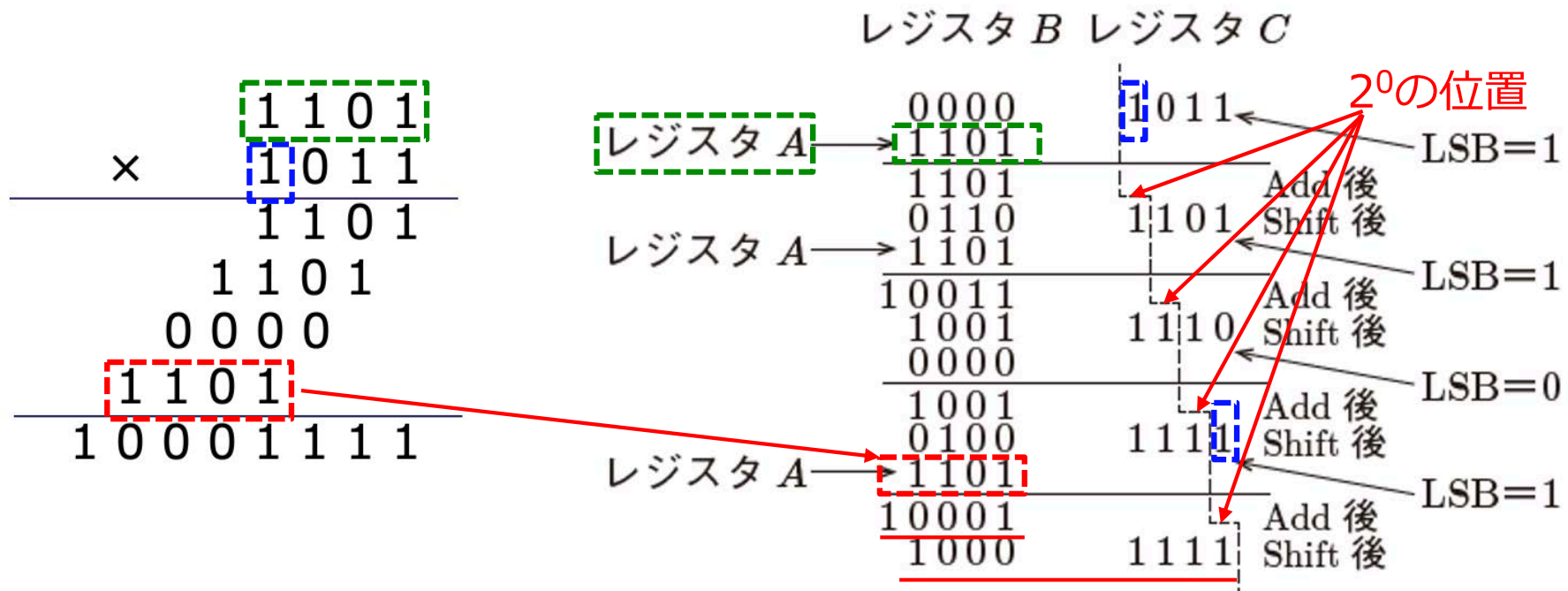


解図 15・2

$N=1101$, $M=1011$ の積 $K=10001111$ を計算するアルゴリズムの概略

二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ
 - 2進数の掛け算 ($1101 * 1011 = 10001111 = 13 * 11 = 143$)

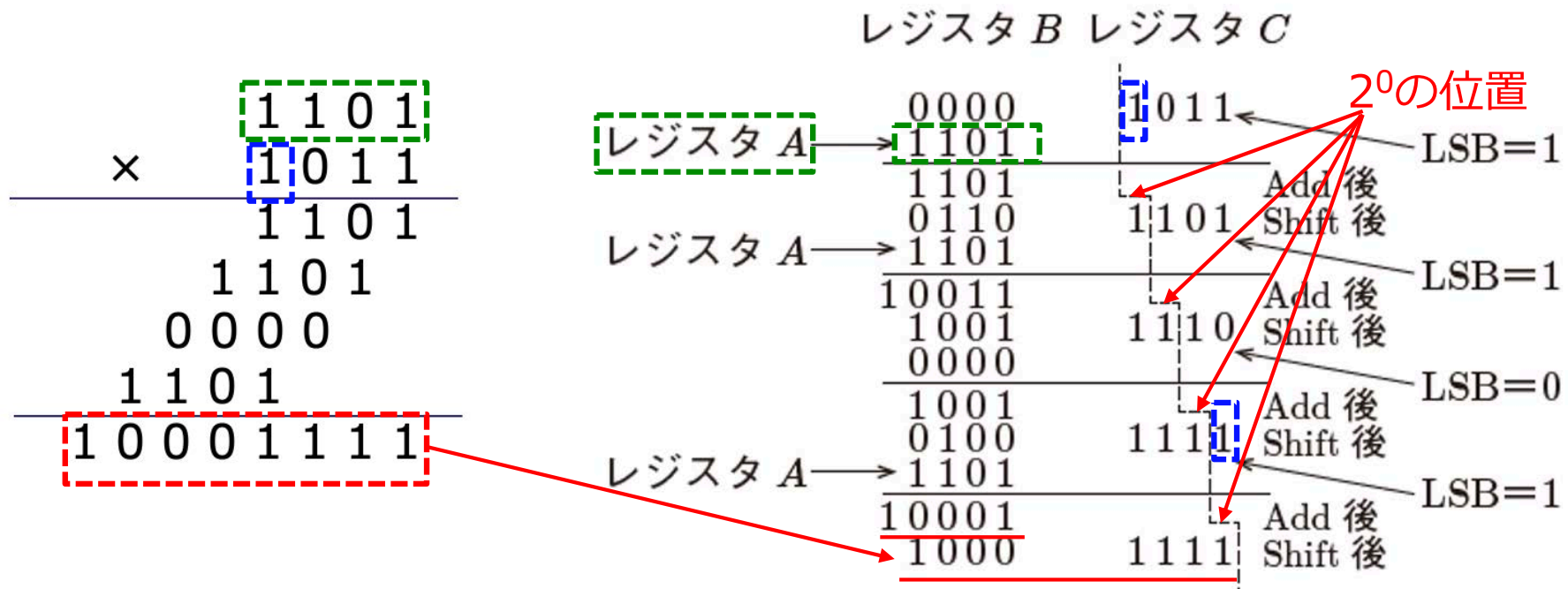


解図 15・2

$N=1101$, $M=1011$ の積 $K=10001111$ を計算するアルゴリズムの概略

二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ
 - 2進数の掛け算 ($1101 * 1011 = 10001111 = 13 * 11 = 143$)



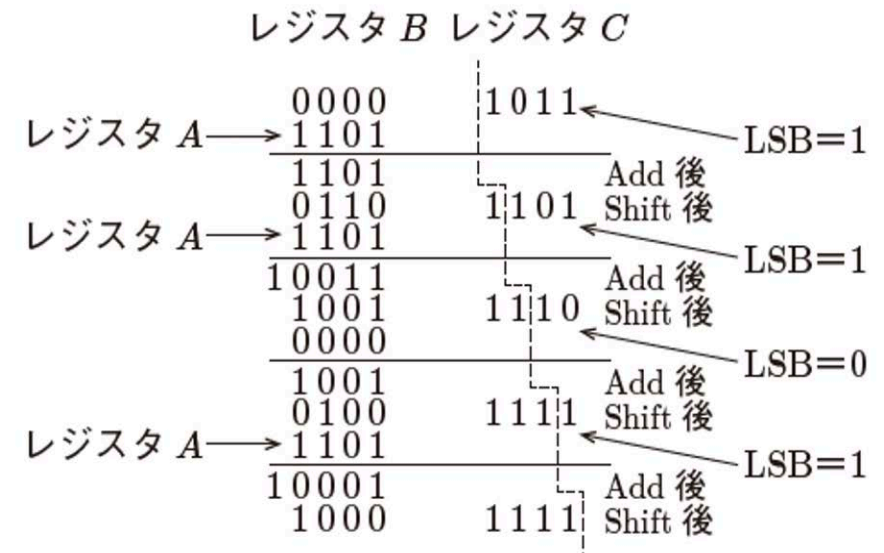
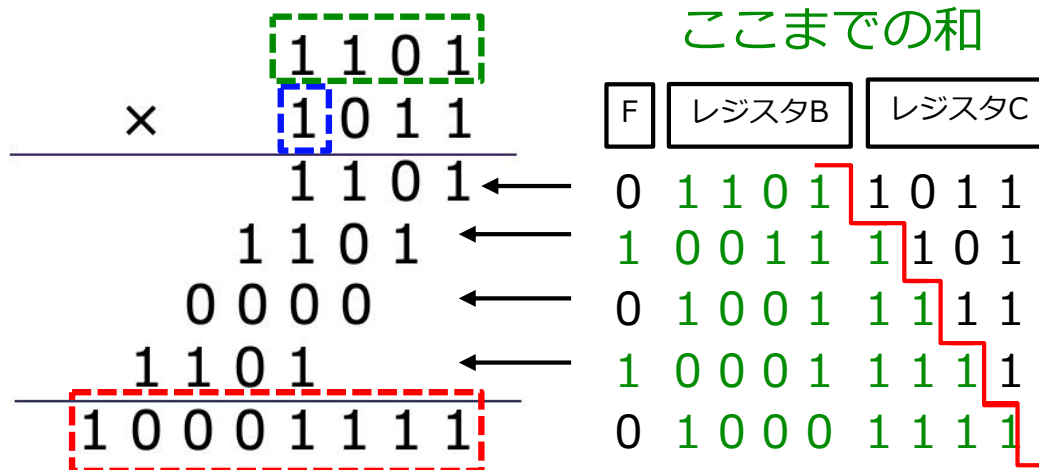
解図 15・2

$N=1101$, $M=1011$ の積 $K=10001111$ を計算するアルゴリズムの概略



二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ
 - 2進数の掛け算 ($1101 * 1011 = 10001111 = 13 * 11 = 143$)

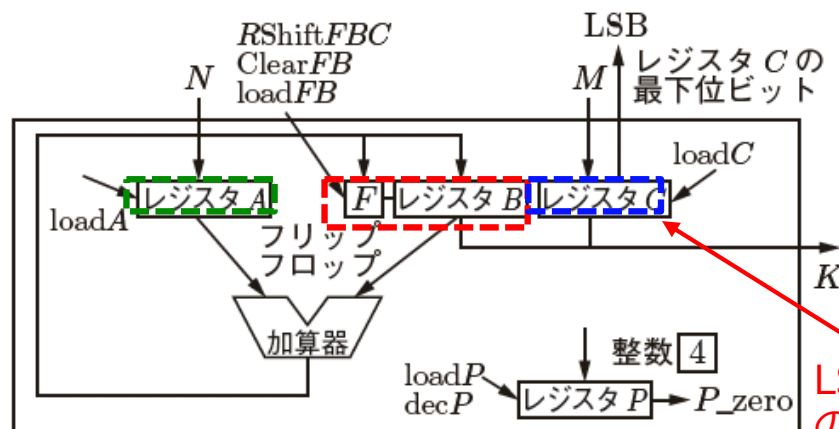


- A+B の和をレジスタ F-B に代入
- F-B-C は 9 ビットのレジスタとして扱い、右に1ビットシフトできる

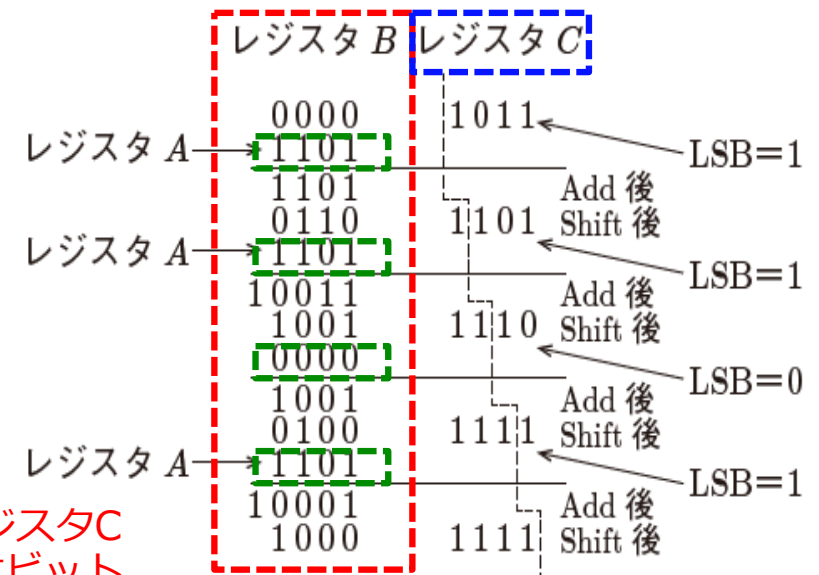


二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ
 - 2進数の掛け算 ($1101 * 1011 = 10001111 = 13 * 11 = 143$)

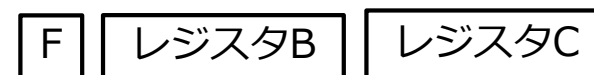


LSB: レジスタ C
の最下位ビット



解図 15・1

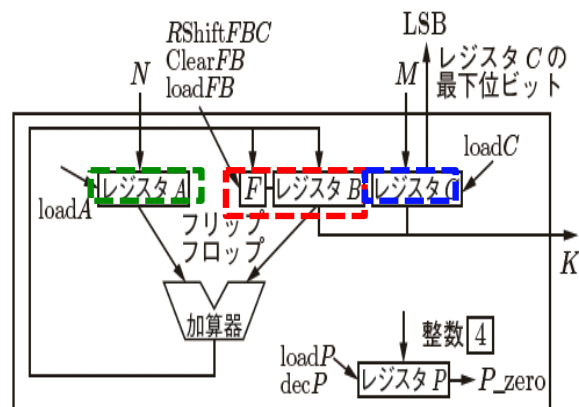
非負整数 N, M の積 K を求める同期式順序回路



- $A+B$ の和をレジスタ F-B に代入
- F-B-C は 9 ビットのレジスタとして扱い、右に 1 ビットシフトできる

二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ
 - 2進数の掛け算 ($1101 * 1011 = 10001111 = 13 * 11 = 143$)



解図 15・1 非負整数 N, M の積 K を求める同期式順序回路

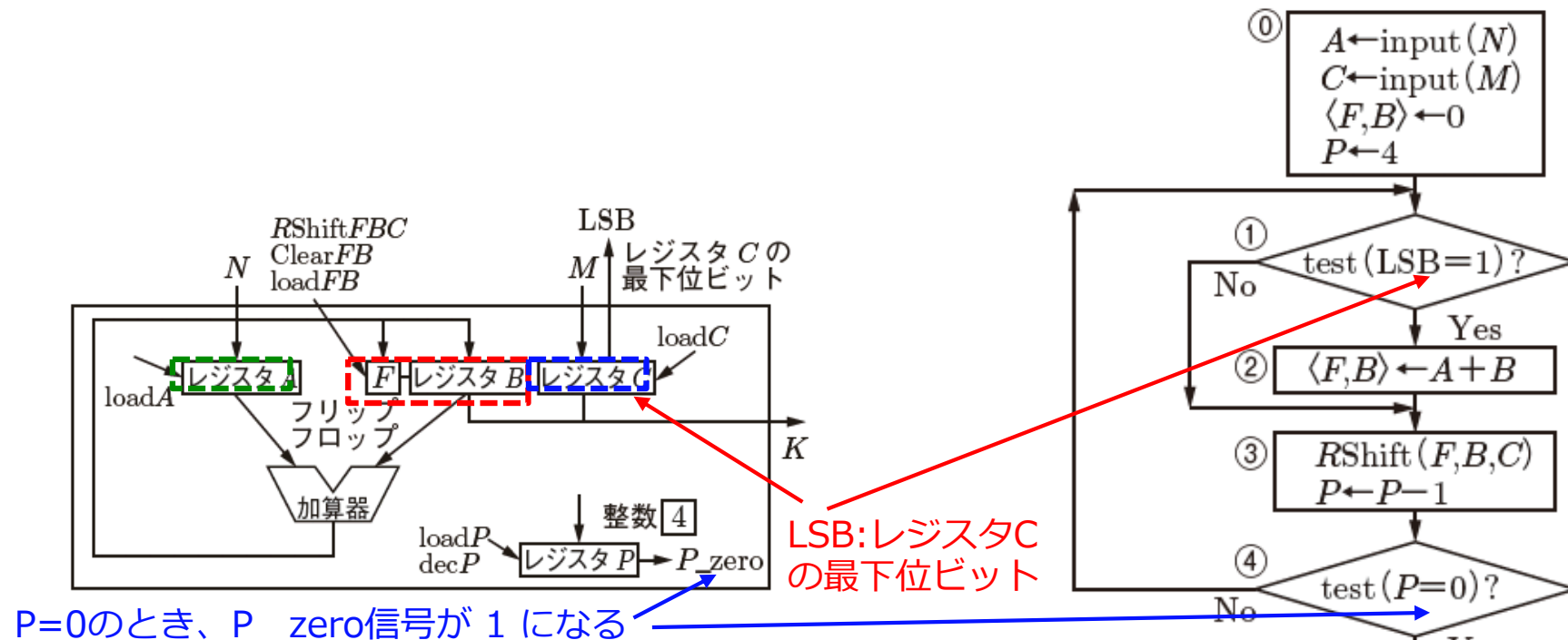
- (イ) $loadA=1$ のとき 外部入力 N をレジスタ A に格納する ($A \leftarrow input(N)$ と書く)
 $loadA=0$ のとき レジスタ A の値は不変
- (ロ) $RShiftFBC=1$, $ClearFB=0$, $loadFB=0$, $loadC=0$ のとき
 F, B, C の値を 1 ビット右にシフトする ($RShift(F, B, C)$ と書く)
 $RShiftFBC=0$, $ClearFB=1$, $loadFB=0$ のとき F, B の値を 0 にする ($\langle F, B \rangle \leftarrow 0$ と書く)
 $RShiftFBC=0$, $ClearFB=0$, $loadFB=1$ のとき 加算器の出力 ($A+B$) を $\langle F, B \rangle$ に格納する
 ($\langle F, B \rangle \leftarrow A+B$ と書く)
- $RShiftFBC=0$, $loadC=1$ のとき
 外部入力 M をレジスタ C に格納する ($C \leftarrow input(M)$ と書く)
 $RShiftFBC=0$, $ClearFB=0$, $loadFB=0$, $loadC=0$ のとき F, B, C の値は不変
- (ハ) $loadP=1$, $decP=0$ のとき 整数 4 をレジスタ P に格納する ($P \leftarrow 4$ と書く)
 $loadP=0$, $decP=1$ のとき レジスタ P の値を 1 減らす ($P \leftarrow P-1$ と書く)
 $loadP=0$, $decP=0$ のとき レジスタ P の値は不変
- (ニ) $P=0$ が真 のときかつそのときのみ $P_zero=1$ ($test(P=0)?$ と書く)
- (ホ) レジスタ C の最下位ビットが 1
 のときかつそのときのみ $LSB=1$ ($test(LSB=1)?$ と書く)





二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ
 - 2進数の掛け算 ($1101 * 1011 = 10001111 = 13 * 11 = 143$)



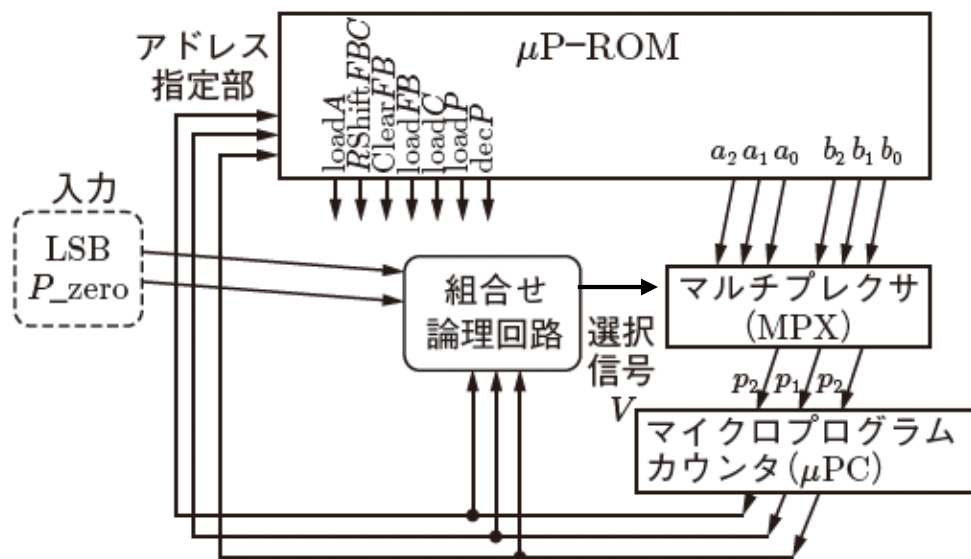
解図 15・1 非負整数 N, M の積 K を求める同期式順序回路

解図 15・4 非負整数 N, M の積 K を求めるアルゴリズムのフローチャート



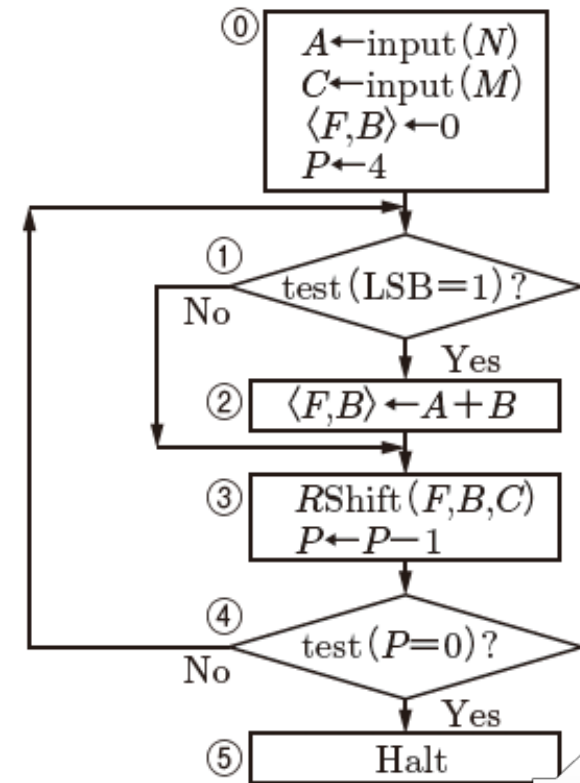
二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ
 - 2進数の掛け算 ($1101 * 1011 = 10001111 = 13 * 11 = 143$)



解図 15・5 非負整数 N, M の積 K を求める同期式順序回路の制御部

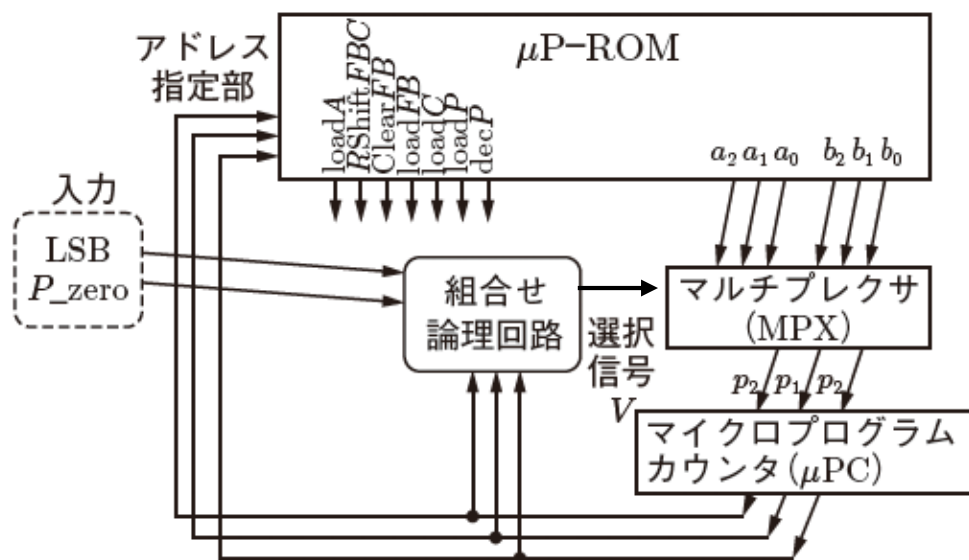
$$V = (\mu PC = (0, 0, 1) \wedge \text{LSB} = 1) \vee (\mu PC = (1, 0, 0) \wedge P = 0) \\ = (\neg p_2 \wedge \neg p_1 \wedge p_0 \wedge \text{LSB}) \vee (p_2 \wedge \neg p_1 \wedge \neg p_0 \wedge P_zero)$$



解図 15・4 非負整数 N, M の積 K を求めるアルゴリズムのフローチャート

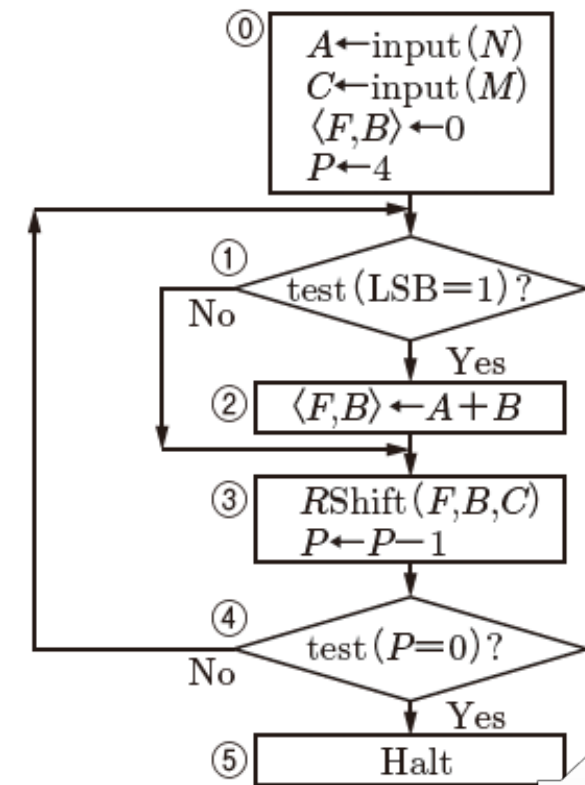
二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ
 - 2進数の掛け算 ($1101 * 1011 = 10001111 = 13 * 11 = 143$)



$\text{loadA} = (\mu\text{PC}=(0,0,0)) = \neg p_2 \wedge \neg p_1 \wedge \neg p_0$
 $\text{RShiftFBC} = (\mu\text{PC}=(0,1,1)) = \neg p_2 \wedge p_1 \wedge p_0$
 $\text{ClearFB} = (\mu\text{PC}=(0,0,0)) = \neg p_2 \wedge \neg p_1 \wedge \neg p_0$
 $\text{loadFB} = (\mu\text{PC}=(0,1,0)) = \neg p_2 \wedge p_1 \wedge \neg p_0$
 $\text{loadC} = (\mu\text{PC}=(0,0,0)) = \neg p_2 \wedge \neg p_1 \wedge \neg p_0$

$\text{loadP} = (\mu\text{PC}=(0,0,0)) = \neg p_2 \wedge \neg p_1 \wedge \neg p_0$
 $\text{decP} = (\mu\text{PC}=(0,1,1)) = \neg p_2 \wedge p_1 \wedge p_0$

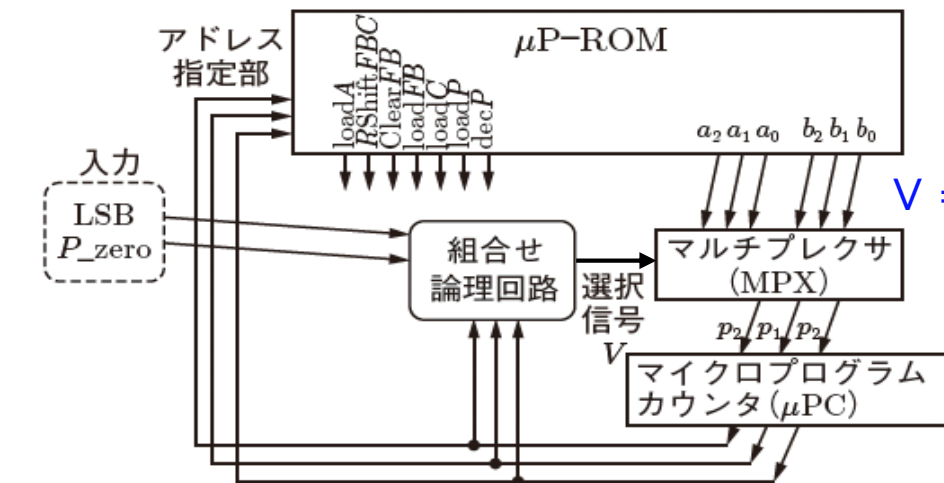




OSAKA UNIVERSITY

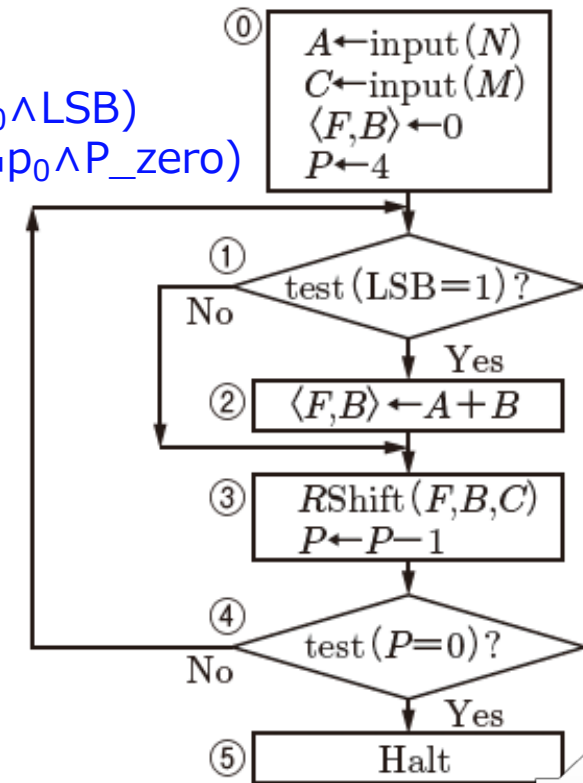
二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ



$$V = (\neg p_2 \wedge \neg p_1 \wedge p_0 \wedge \text{LSB}) \vee (p_2 \wedge \neg p_1 \wedge \neg p_0 \wedge P_zero)$$

マイクロ命令	loadA	RShiftFBC	clearFB	loadFB	loadC	loadP	decP	a ₂	a ₁	a ₀	b ₂	b ₁	b ₀
番地 0 (①)	1	0	1	0	1	1	0	0	0	1	X	X	X
番地 1 (②)	0	0	0	0	0	0	0	0	1	1	0	1	0
番地 2 (③)	0	0	0	1	0	0	0	0	1	1	X	X	X
番地 3 (④)	0	1	0	0	0	0	1	1	0	0	X	X	X
番地 4 (⑤)	0	0	0	0	0	0	0	0	0	1	1	0	1
番地 5 (⑥)	0	0	0	0	0	0	0	1	0	1	X	X	X



解図 15・6

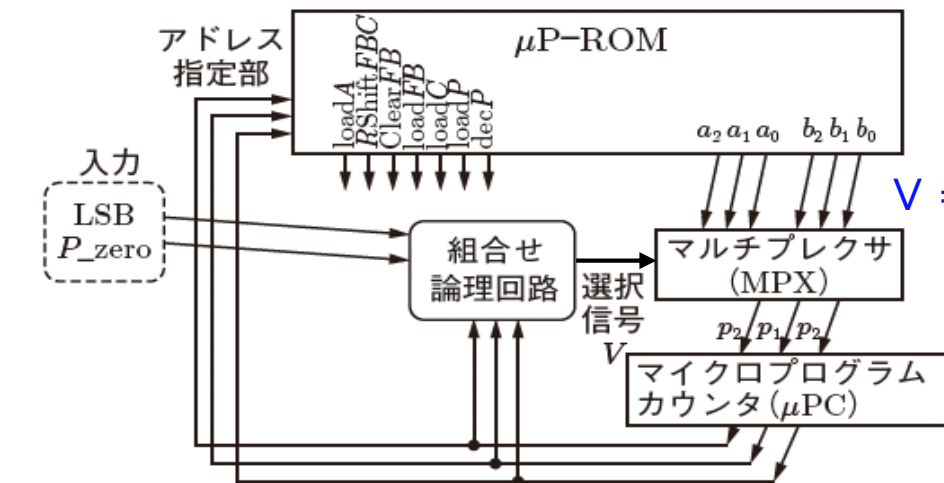
N, M の積 K を求める同期式順序回路の制御部の μP-ROM の内容



OSAKA UNIVERSITY

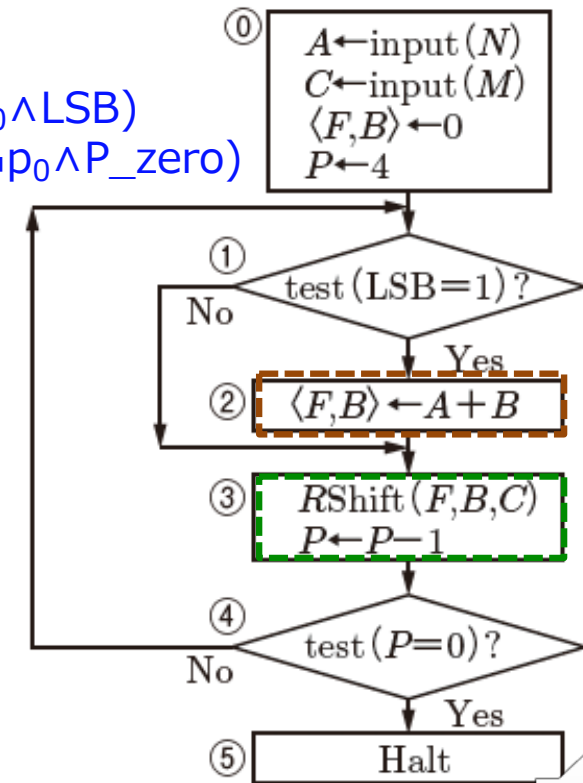
二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ



$$V = (\neg p_2 \wedge \neg p_1 \wedge p_0 \wedge \text{LSB}) \vee (p_2 \wedge \neg p_1 \wedge \neg p_0 \wedge P_zero)$$

マイクロ命令	loadA	RShiftFBC	clearFB	loadFB	loadC	loadP	decP	a ₂	a ₁	a ₀	b ₂	b ₁	b ₀
番地 0 (①)	1	0	1	0	1	1	0	0	0	1	X	X	X
番地 1 (①)	0	0	0	0	0	0	0	0	1	1	0	1	0
番地 2 (②)	0	0	0	1	0	0	0	0	1	1	X	X	X
番地 3 (③)	0	1	0	0	0	0	1	1	0	0	X	X	X
番地 4 (④)	0	0	0	0	0	0	0	0	0	1	1	0	1
番地 5 (⑤)	0	0	0	0	0	0	0	1	0	1	X	X	X



解図 15・6

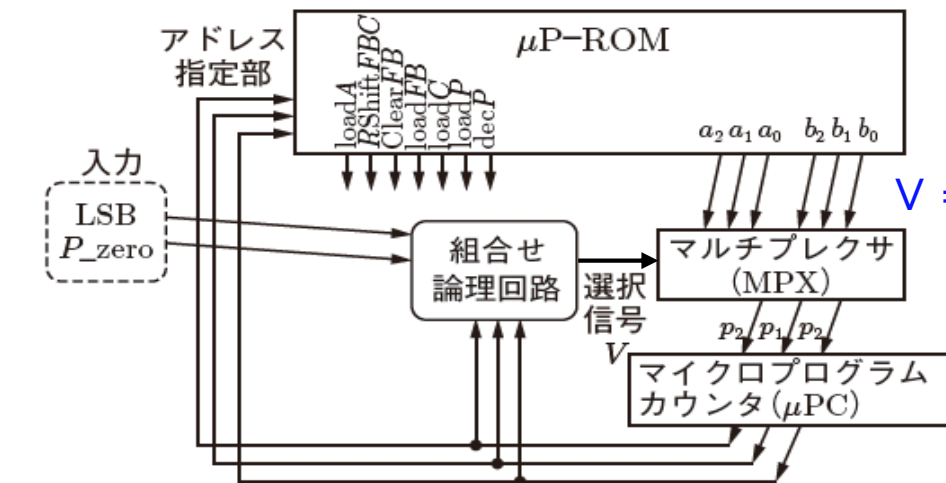
N, M の積 K を求める同期式順序回路の制御部の μP-ROM の内容



OSAKA UNIVERSITY

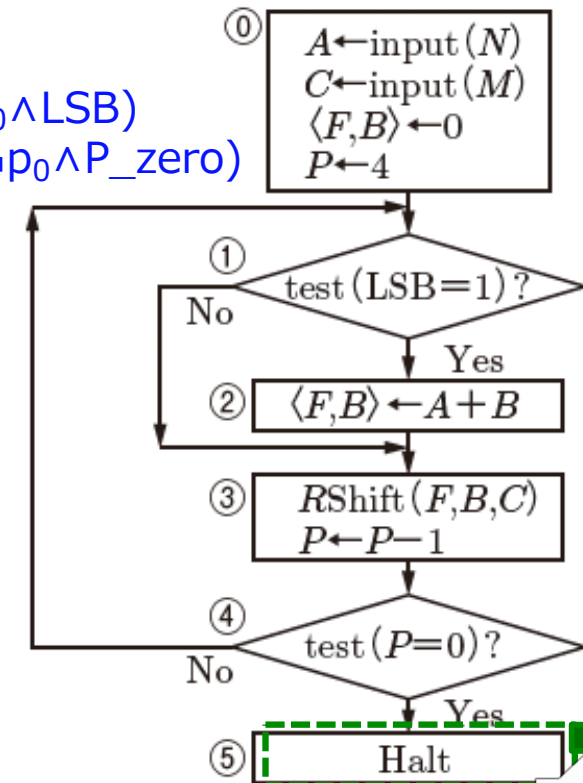
二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ



$$V = (\neg p_2 \wedge \neg p_1 \wedge p_0 \wedge \text{LSB}) \vee (p_2 \wedge \neg p_1 \wedge \neg p_0 \wedge P_zero)$$

マイクロ命令	loadA	RShiftFBC	clearFB	loadFB	loadC	loadP	decP	a ₂	a ₁	a ₀	b ₂	b ₁	b ₀
番地 0 (①)	1	0	1	0	1	1	0	0	0	1	X	X	X
番地 1 (②)	0	0	0	0	0	0	0	0	1	1	0	1	0
番地 2 (③)	0	0	0	1	0	0	0	0	1	1	X	X	X
番地 3 (④)	0	1	0	0	0	0	1	1	0	0	X	X	X
番地 4 (⑤)	0	0	0	0	0	0	0	0	0	1	1	0	1
番地 5 (⑥)	0	0	0	0	0	0	0	1	0	1	X	X	X



解図 15・6

N, M の積 K を求める同期式順序回路の制御部の μP-ROM の内容



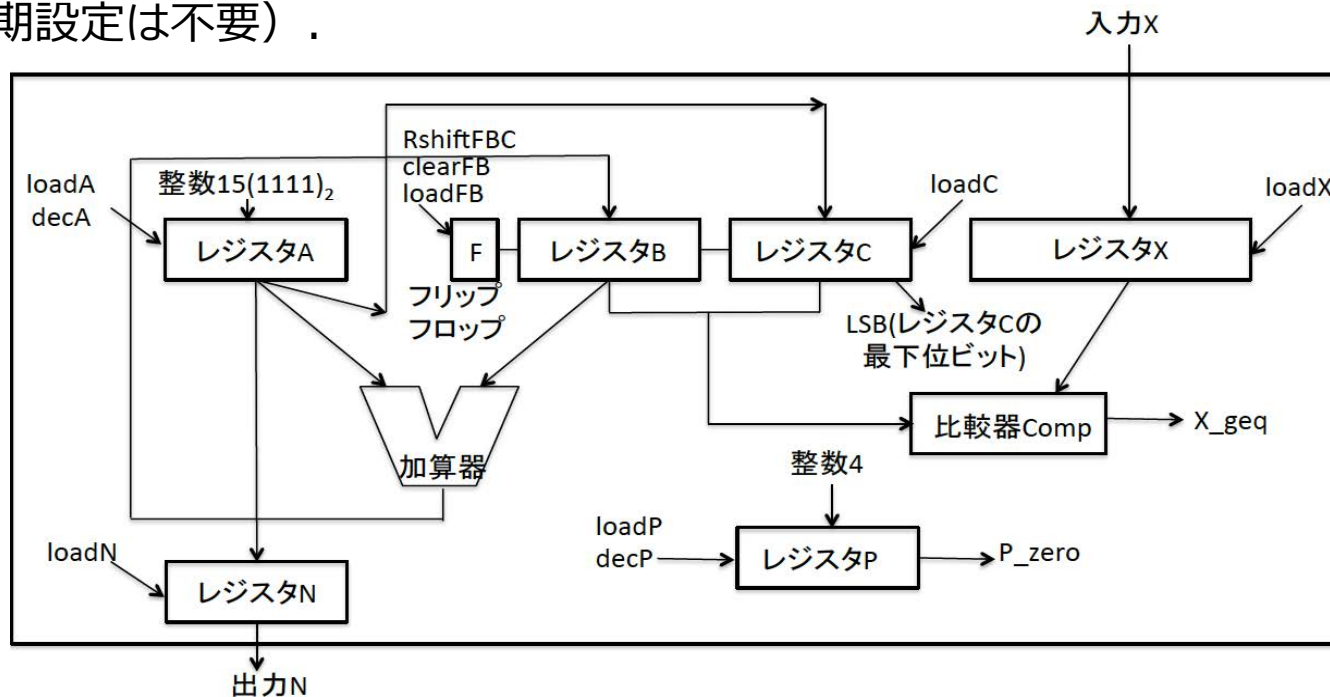
本日のレポート課題 (演習問題 2)



演習問題 2

与えられた数 n の平方根 \sqrt{n} を求める同期式順序回路を設計せよ

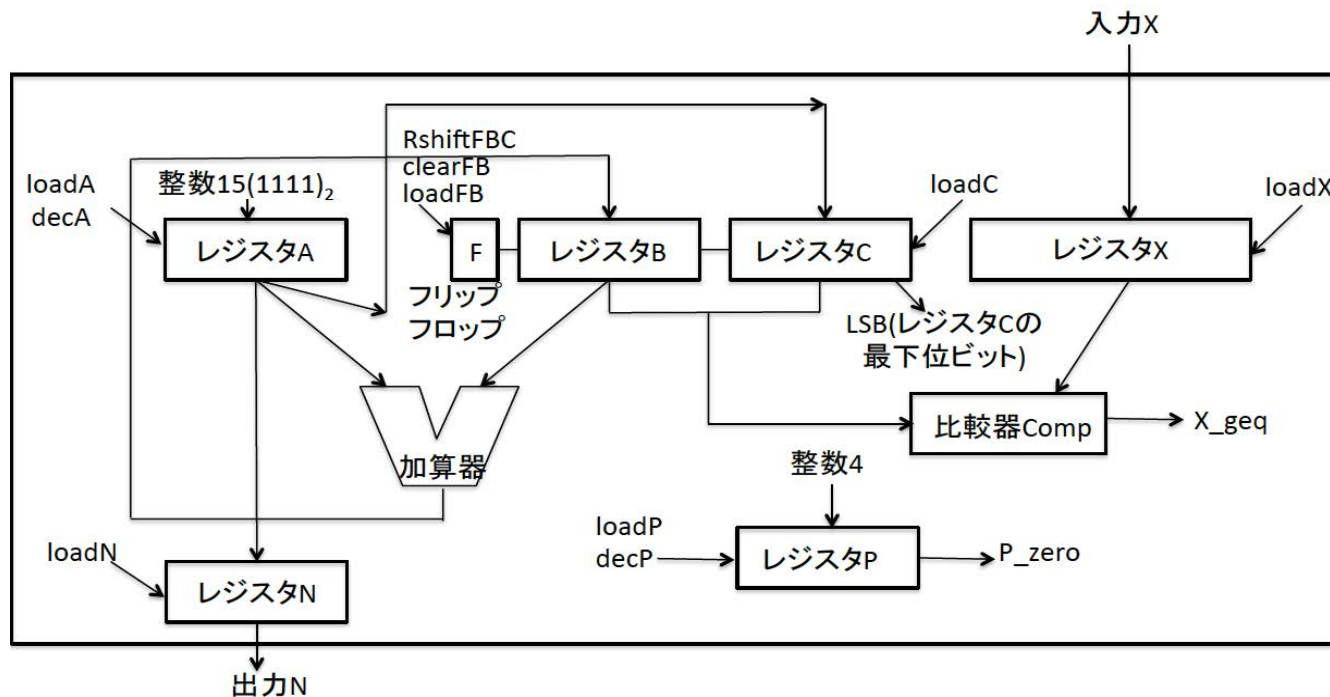
- 下図のような回路を用いて、外部から入力された 8 ビットの非負整数 X に対して、 X の平方根の近似値 ($(N+1)^2 > X \geq N^2$ となる 4 ビットの非負整数 N) を求める同期式順序回路を作成したい。図のレジスタA とレジスタC に最初 4 ビットの最大値 $15(1111)_2$ を N の値としてセットし、レジスタP に整数値 4 をセットする。レジスタA とレジスタC の積 N^2 を計算し、その値とレジスタX の値を比較する。レジスタX の値が N^2 以上 ($X \geq N^2$) になれば比較器 Comp の X_geq 信号が 1 になり、その時のレジスタA の値を X の平方根の近似値としてレジスタN にセットする (レジスタN の初期設定は不要)。



演習問題 2

与えられた数 n の平方根 \sqrt{n} を求める同期式順序回路を設計せよ

- N^2 の計算の際に 1 ビットのフリップフロップF とレジスタB を作業用レジスタとして利用する（初期設定必要）. 加算器はレジスタA とレジスタBの和を計算し, その値をレジスタB に格納する. レジスタA とレジスタB の和が 5 ビットになった場合は 5 ビット目の内容がフリップフロップF に格納される. フリップフロップF とレジスタB, レジスタC はこの順に接続され, RshiftFBC信号 を 1 にすると右に 1 ビットシフトされ, 最上位のフリップフロップF に 0 がセットされる. レジスタC の最下位ビットが LSB信号 として出力される.

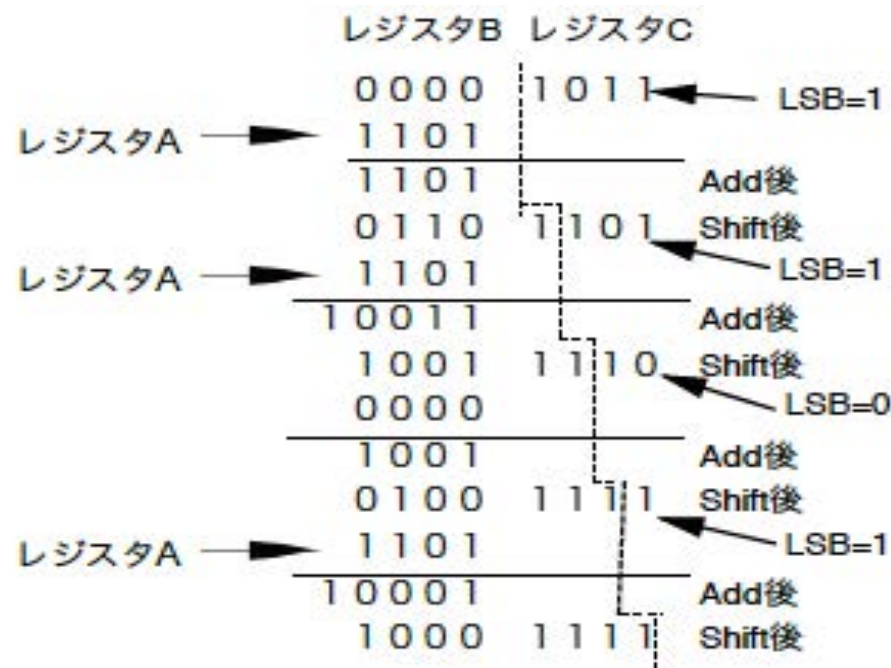




演習問題 2

与えられた数 n の平方根 \sqrt{n} を求める同期式順序回路を設計せよ

- レジスタAとレジスタCの積の計算は、下図のようにAdd（加算）、Shift（右に1ビットシフト）を繰り返して行う（LSBが1 のときレジスタAの値をレジスタBに加算する）。最終的にレジスタAとレジスタCの積がレジスタB、レジスタCに格納される（レジスタBが上位4ビット、レジスタCが下位4ビットになる）。これは演習問題 1 で説明したかけ算のアルゴリズムと同じです。





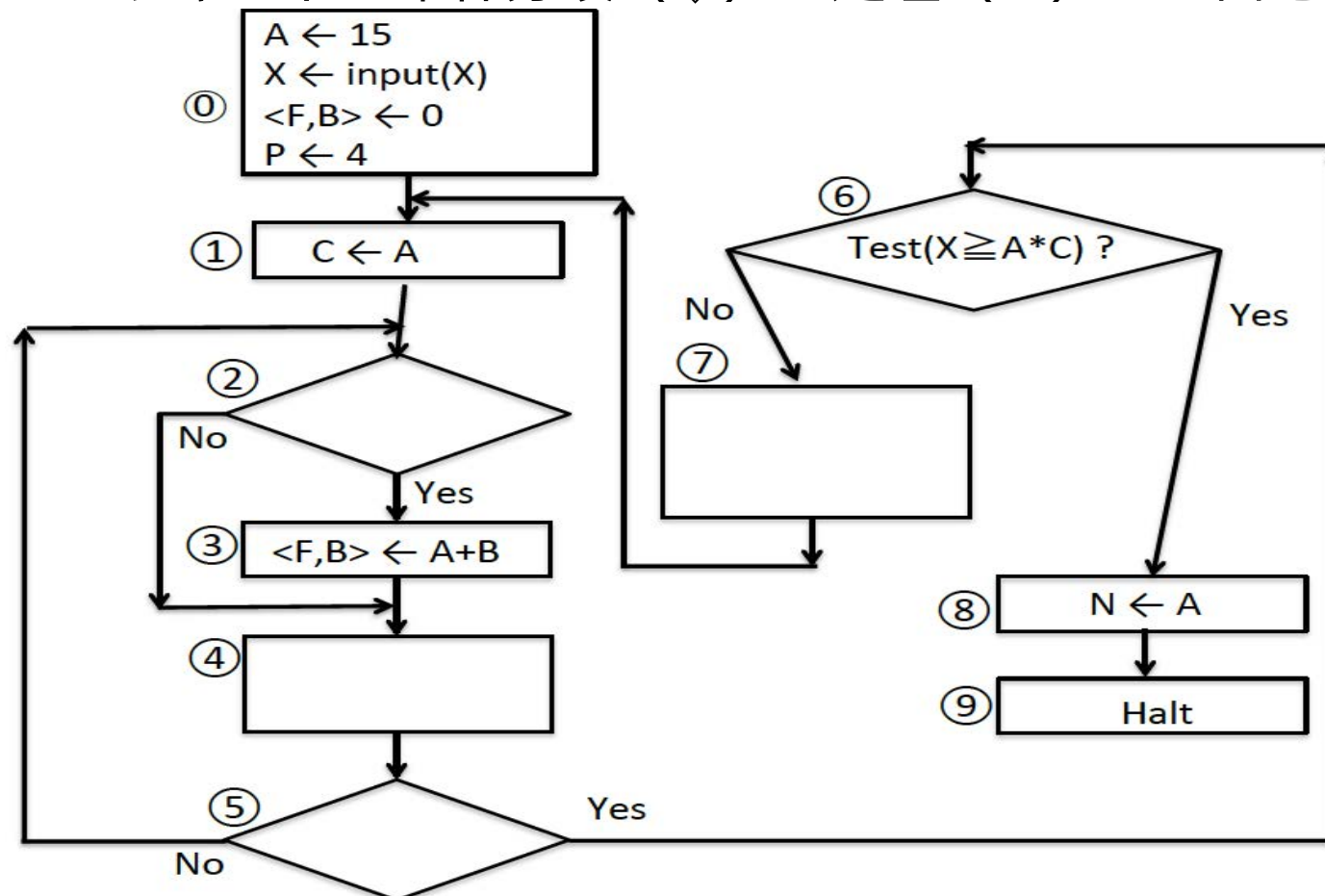
演習問題 2

- 各部品は次のように動作するものと仮定する。
 - (a) loadA=1, decA=0のとき 4ビットの最大値15(1111_2)をレジスタAに格納 ($A \leftarrow 15$)
 loadA=0, decA=1のとき レジスタAの値を1減らす ($A \leftarrow A-1$)
 loadA=0, decA=0のとき レジスタAの値は不変
 - (b) RshiftFBC=1, clearFB=0, loadFB=0, loadC=0, decC=0 のとき
 F,B,Cの値を1ビット右にシフトする (Rshift(F,B,C) と書く)
 RshiftFBC=0, clearFB=1, loadFB=0 のときF,Bの値を0にする ($\langle F, B \rangle \leftarrow 0$)
 RshiftFBC=0, clearFB=0, loadFB=1 のとき(A+B)を $\langle F, B \rangle$ に格納 ($\langle F, B \rangle \leftarrow A+B$)
 RshiftFBC=0, clearFB=0, loadFB=0, loadC=0, decC=0 のときF,B,Cの値は不変
 - (c) loadC=1のとき レジスタAの値をレジスタCに格納する ($C \leftarrow A$)
 loadC=0のとき レジスタCの値は不変
 - (d) loadX=1のとき 8ビットの入力XをレジスタXに格納する ($X \leftarrow \text{input}(X)$)
 loadX=0のとき レジスタXの値は不変
 - (e) loadP=1, decP=0 のとき 整数4をレジスタPに格納する ($P \leftarrow 4$)
 loadP=0, decP=1 のとき レジスタPの値を1減らす ($P \leftarrow P-1$)
 loadP=0, decP=0 のとき レジスタPの値は不変
 - (f) loadN=1のとき レジスタAの値をレジスタNに格納する ($N \leftarrow A$)
 loadN=0のとき レジスタNの値は不変
 - (g) P=0 が真のときかつそのときのみ P_zero=1 (test(P=0)? と書く)
 - (h) $X \geq A * C (= \langle B, C \rangle)$ が真のときかつそのときのみ X_geq=1 (test($X \geq A * C$)? と書く)
 - (i) レジスタCの最下位ビットが1のとき かつそのときのみ LSB=1 (test(LSB=1)? と書く)



演習問題 2

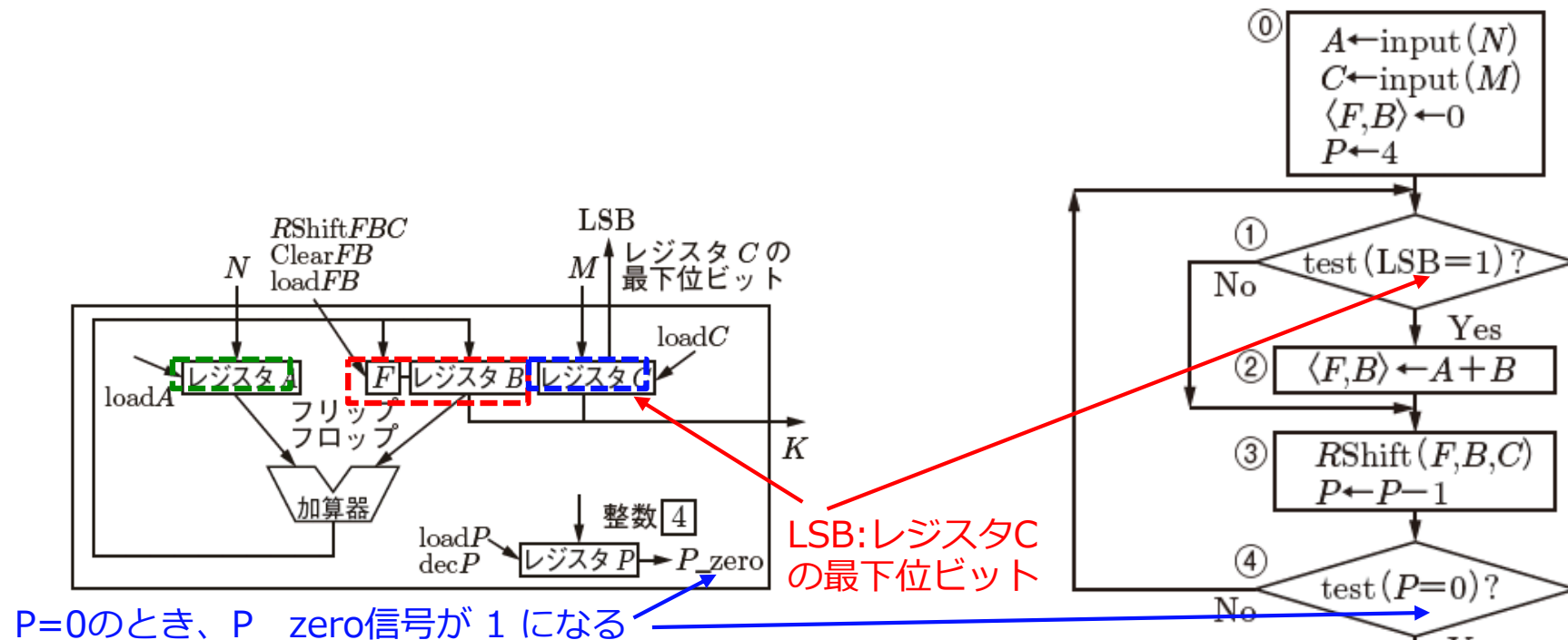
- (a)～(i)の動作仕様をもとに，非負整数 X の平方根の近似値 $((N+1)^2 > X \geq N^2$ となる N) を求めるアルゴリズムのフローチャートを考え，下図のフローチャートの条件分岐 (◇) と処理 (□) の空白を埋めよ.





二つの整数の積を計算する 同期式順序回路

- 電卓の回路や自動販売機の制御回路と同様の方法で、二つの整数の積を計算する同期式順序回路を設計せよ
 - 2進数の掛け算 ($1101 * 1011 = 10001111 = 13 * 11 = 143$)



解図 15・1 非負整数 N, M の積 K を求める同期式順序回路

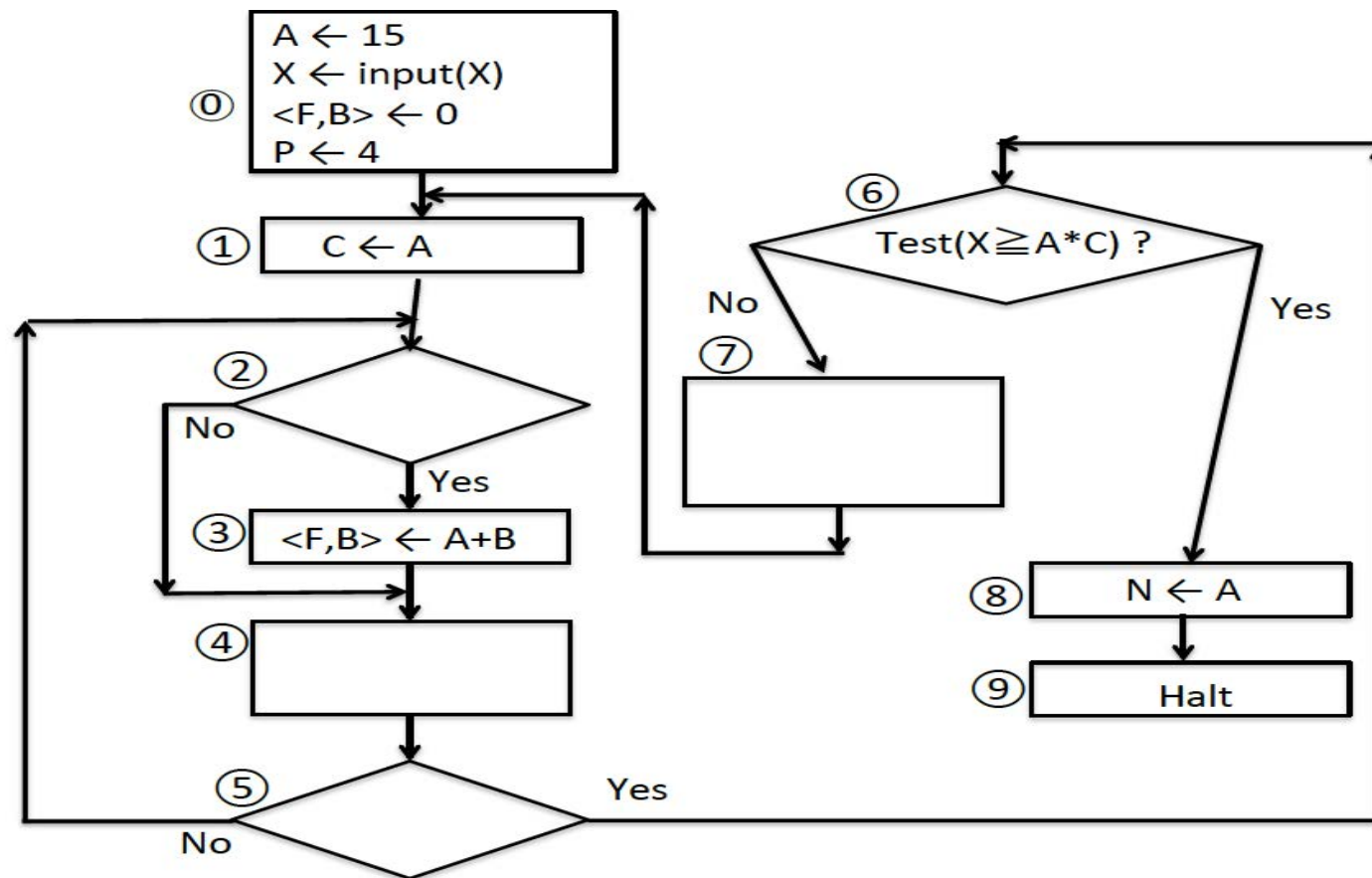
解図 15・4 非負整数 N, M の積 K を求めるアルゴリズムのフローチャート



演習問題 2

(問題 1)

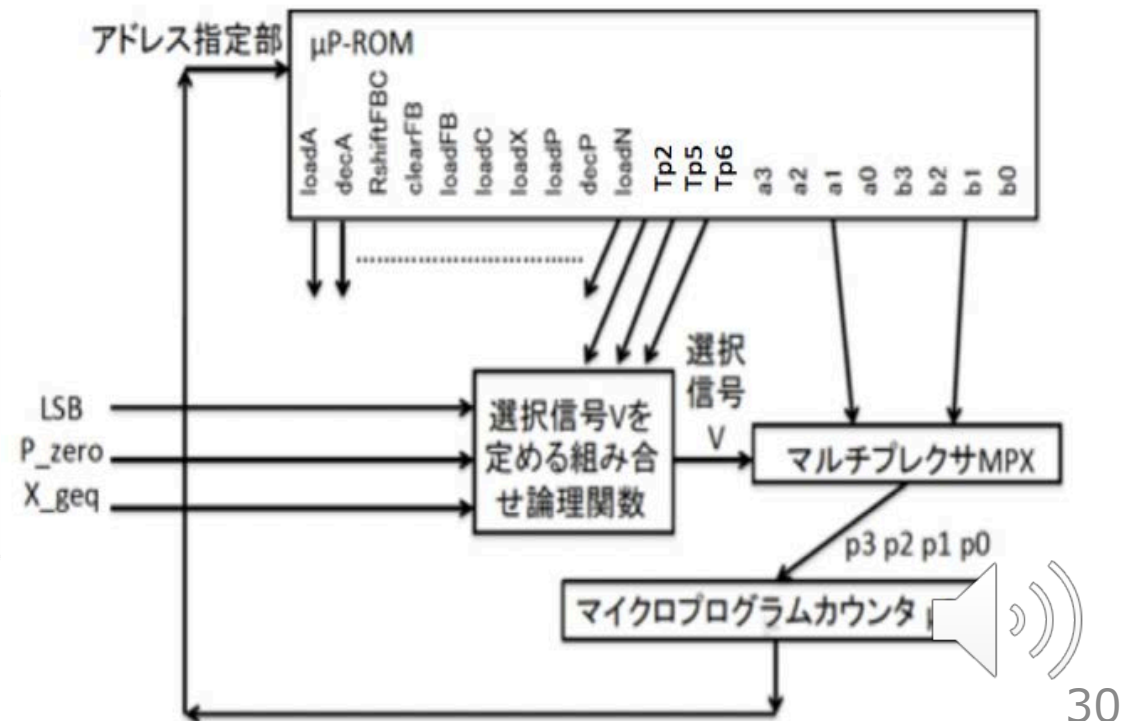
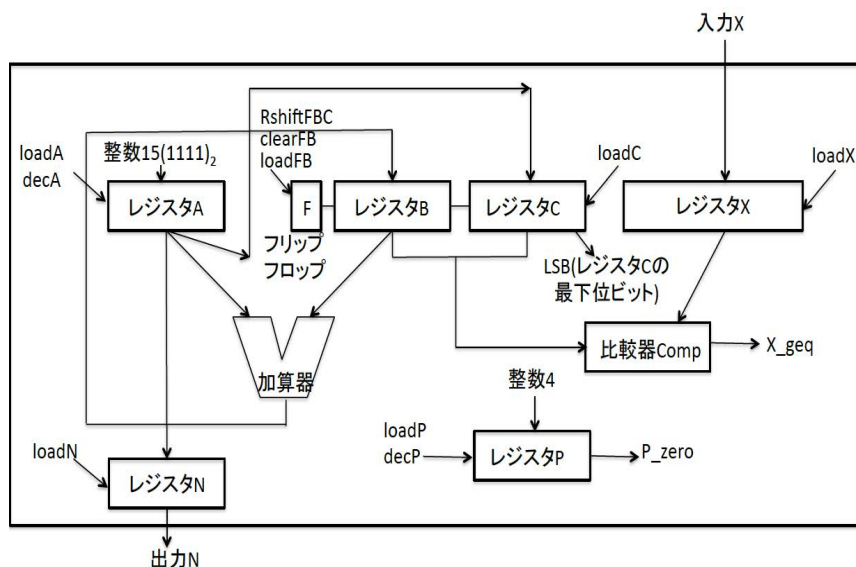
- (a)～(i)の動作仕様をもとに，非負整数 X の平方根の近似値
 $((N+1)^2 > X \geq N^2$ となる N)を求めるアルゴリズムのフローチャートを考え，下図のフローチャートの空白を埋めよ.





演習問題 2

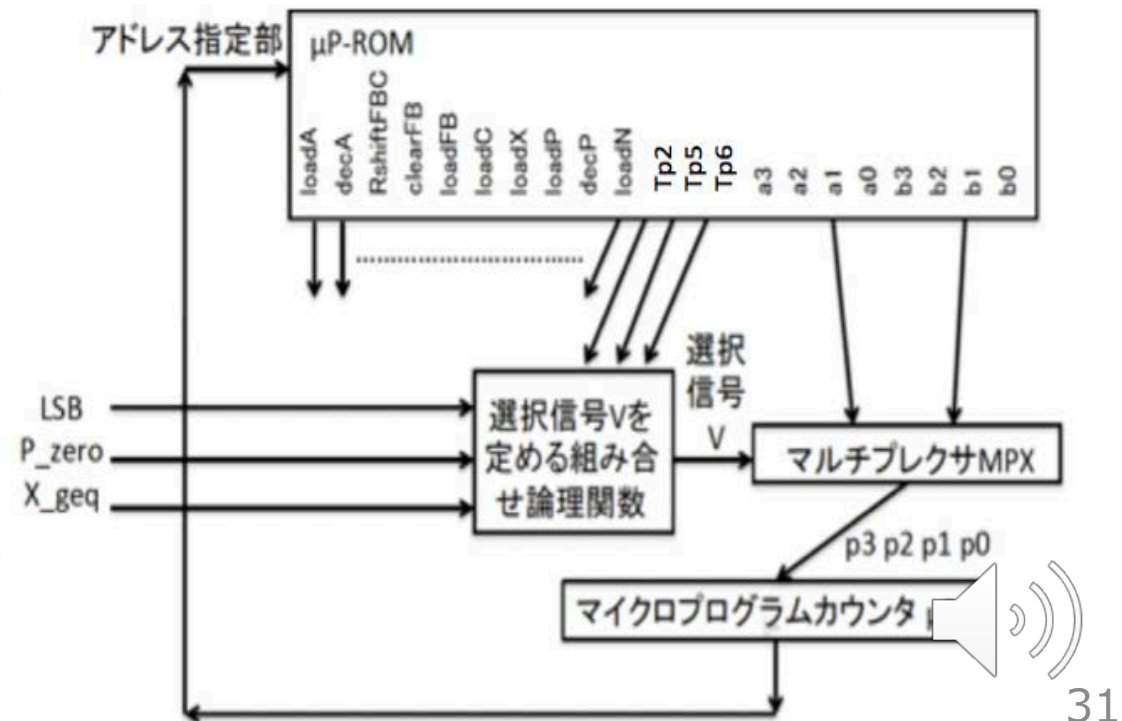
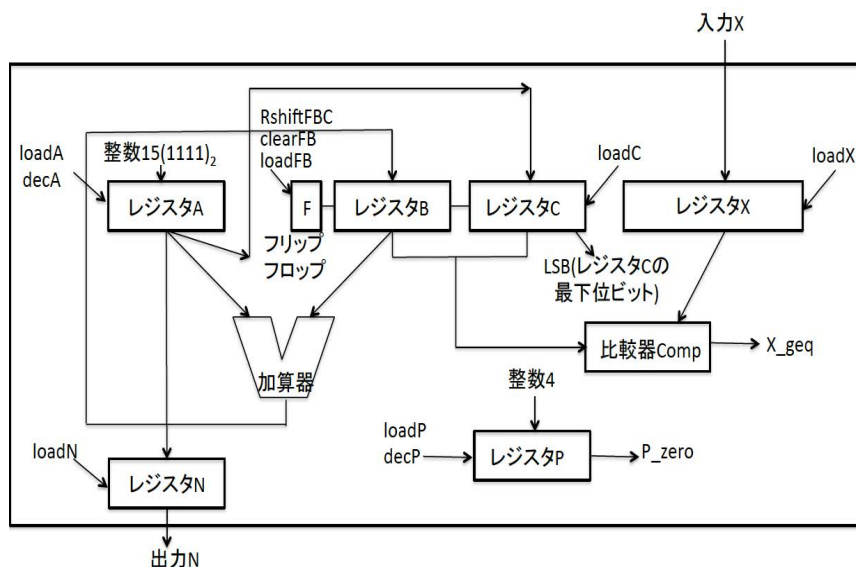
- 左下図の順序回路の制御部を右下図のような制御回路で実現したい。
右下図のマイクロプログラムカウンタ μPC の値 $p_3p_2p_1p_0$ (p_0 が最下位ビット) は上記で作成したフローチャートの条件分岐や処理に対応するアドレスを表す。 μP -ROM の $a_3a_2a_1a_0$, $b_3b_2b_1b_0$ はそれぞれ次に実行するアドレスを表す (a_0, b_0 が最下位ビット)。





演習問題 2

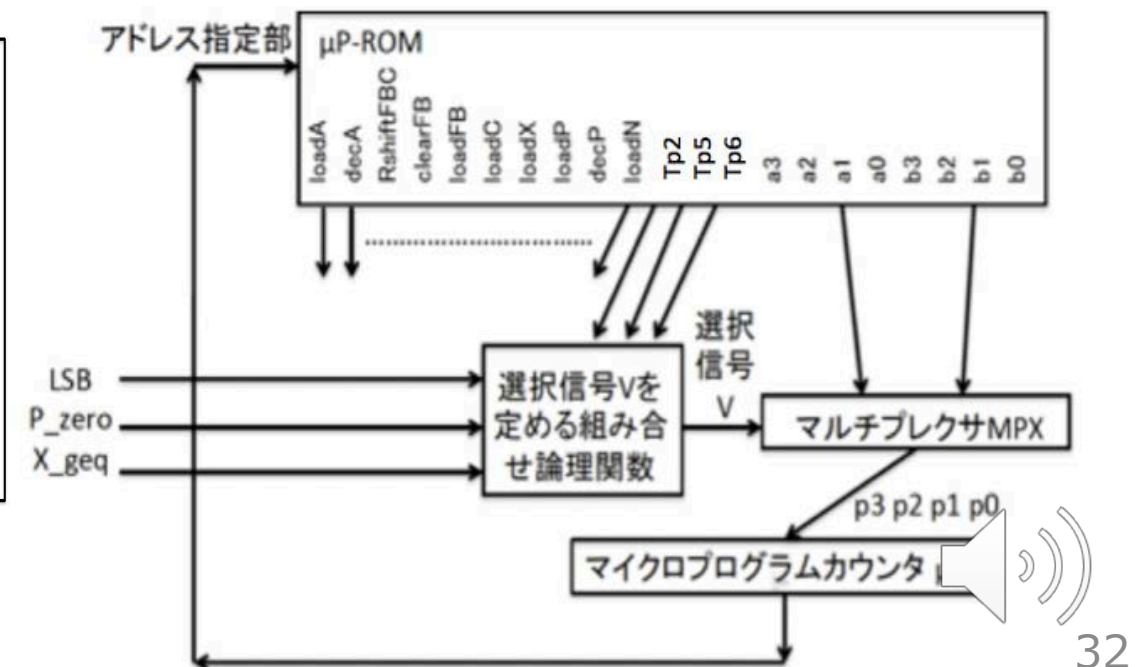
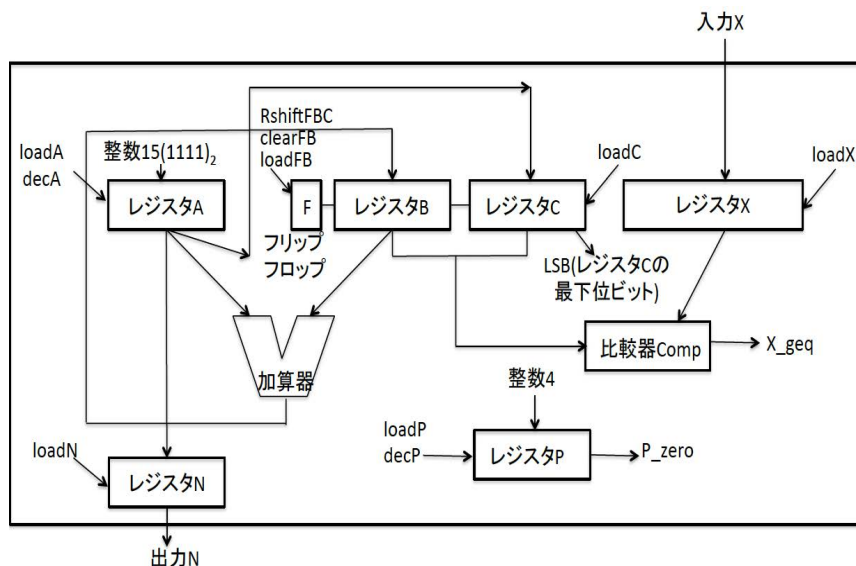
- マルチプレクサ MPX は選択信号 V の値が 0 のとき $a_3a_2a_1a_0$ を出力し, V の値が 1 のとき $b_3b_2b_1b_0$ を出力する. また, 制御信号 loadA, decA, RshiftFBC, clearFB, loadFB, loadC, loadX, loadP, decP, loadN の値は, マイクロプログラムカウンタ μPC の値 p_3, p_2, p_1, p_0 をアドレスとして μP -ROM にその信号値を保存することで実現する.





演習問題 2

- (問題2-1) 制御信号 loadA, decA, RshiftFBC の値をそれぞれ p_3, p_2, p_1, p_0 を引数とする論理式で表せ.
- (問題2-2) 選択信号 V の値を $T_{p2}, T_{p5}, T_{p6}, \text{LSB}, P_zero, X_geq$ の論理式で表せ (条件分岐 ②, ⑤, ⑥ に対応して T_{p2}, T_{p5}, T_{p6} を用いよ).
- (問題2-3) この回路の μ P-ROM の内容とその 0 番地から Z 番地 ($Z \leq 9$) の内容を定めよ. ドント・ケアでよい部分は×印を付けよ.





演習問題 2

- μ P-ROMの内容は下記の表を埋めて下さい
- 解答はCLEにアップした解答用紙に記入して下さい

	p3 p2 p1 p0	loadA	decA	RshiftFBC	clearFB	loadFB	loadC	loadX	loadP	decP	loadN	tp2	tp5	tp6	a3	a2	a1	a0	b3	b2	b1	b0
0 番地	0 0 0 0																					
1 番地	0 0 0 1																					
2 番地	0 0 1 0																					
3 番地	0 0 1 1																					
4 番地	0 1 0 0																					
5 番地	0 1 0 1																					
6 番地	0 1 1 0																					
7 番地	0 1 1 1																					
8 番地	1 0 0 0																					
9 番地	1 0 0 1																					

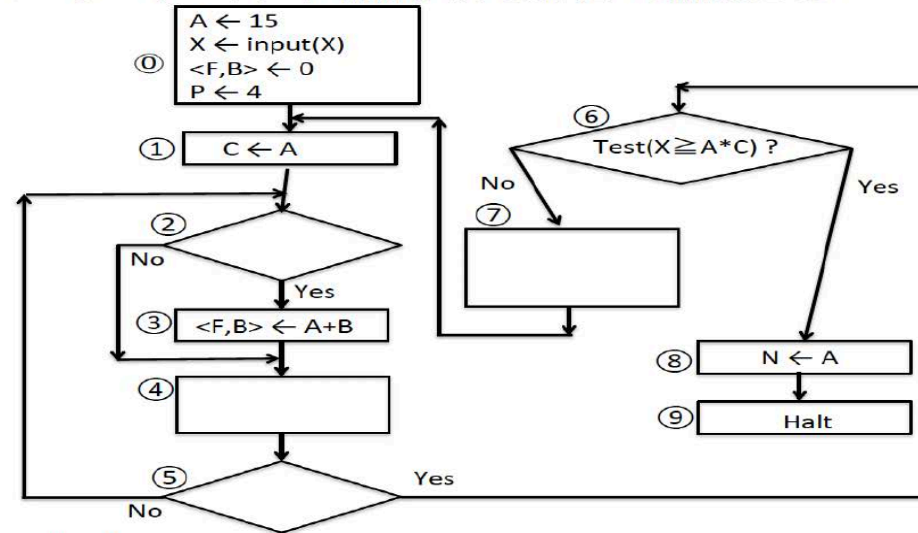




演習問題 2 解答用紙

論理設計演習課題解答用紙

(1) アルゴリズム (フローチャートの条件分岐 (◇) と処理 (□) の空白部分を埋めよ.)



(2-1) 制御信号

loadA =

decA =

RshiftFBC =

(2-2) 選択信号

V =

(2-3) μ P-ROM の内容

	p3	p2	p1	p0	loadA	decA	RshiftFBC	clearFB	loadFB	loadC	loadX	loadP	decP	loadN	TP2	TP5	TP6	a3	a2	a1	a0	b3	b2	b1	b0
0 番地	0	0	0	0																					
1 番地	0	0	0	1																					
2 番地	0	0	1	0																					
3 番地	0	0	1	1																					
4 番地	0	1	0	0																					
5 番地	0	1	0	1																					
6 番地	0	1	1	0																					
7 番地	0	1	1	1																					
8 番地	1	0	0	0																					
9 番地	1	0	0	1																					





かけ算回路と平方根の回路

- かけ算回路
 - 今回紹介したかけ算回路は、加算とシフトでかけ算を実現している。計算機ではかけ算を加算とシフトで実現するのが一般的
- 平方根の回路
 - 今回紹介した平方根の回路の考え方で $\sqrt{2}$ を小数点以下3桁で計算しようとする、2.000から1.999, 1.998, ... と順に小さくしていき、1.414にたどり着くことになる（計算効率が悪い）。
 - $\sqrt{2}$ は1と2の間に答があるので、1と2の midpoint の1.5と予想して、 1.5^2 を計算すると2.25となり2を超えるので、答は1と1.5の間にあると考え、1と1.5の midpoint の1.25の2乗1.5625を計算。 1.25^2 は2より小さいので、答は1.25と1.5の間にある。このような二分法を用いて、順に範囲を狭めていけば上記の方法より効率的に1.414にたどり着く。
 - いずれの場合も平方根の計算はかけ算と減算、大小比較しか使っていない。そのかけ算も加算とシフト、減算、=判定しか使っていない。これはソフトで数値計算する場合も同じ。





11回目の授業終了





授業終了

皆さん
レポート提出してくださいね！