

演算機能回路

第8週

2006/11/20

泉知論

立命館大学 理工学部 電子情報デザイン学科

マルチサイクル演算

multiple-cycle operation

組合せ回路による計算

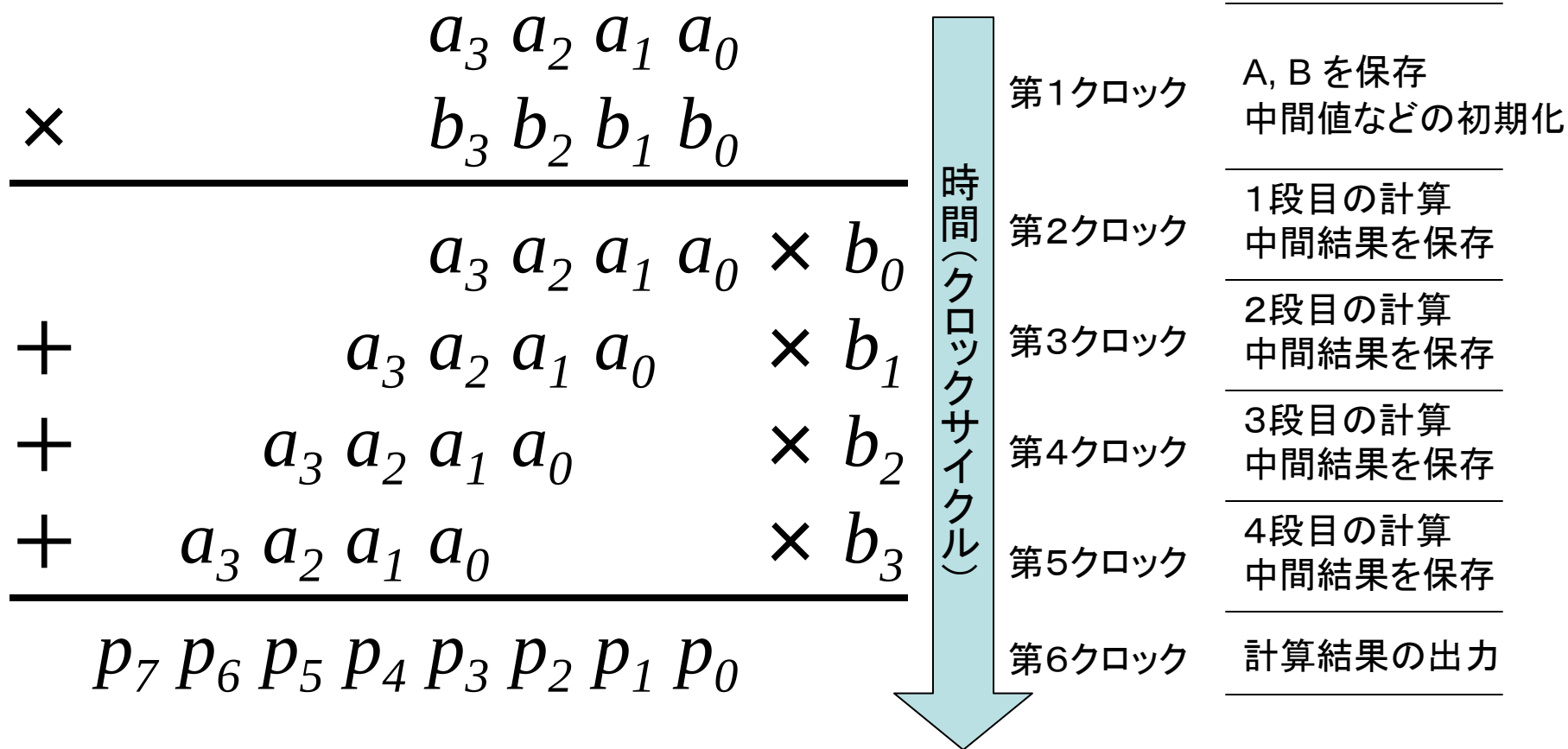
- ・ これまでのモジュールはすべて組合せ回路
 - － 加算器
 - ・ 桁上伝播加算器 CPA
 - ・ 桁上先読加算器 CLA
 - － 乗算器
 - ・ 筆算の単純実装
 - ・ CSA を用いた単純実装
 - ・ Wallace Tree による改良
- ・ 組合せ回路のまま動作周波数を上げる(= 段数を減らす)のにも限界がある

組合せ回路 vs 順序回路

- 組合せ回路で1クロックサイクルで計算
 - 段数が多くなる(=動作周波数が下がる)
- 部分計算への分割→マルチサイクル化
 - それぞれの部分計算を1サイクルで計算
 - 中間値をレジスタに保持
 - 複数サイクルで計算
 - 組合せ回路の段数は減る(=動作周波数を上げられる)
 - しかし、全計算にかかる時間は増加する

乗算の分割例(単純分割)

- 4 bit × 4 bit の例



複数サイクル乗算器の原理と特徴

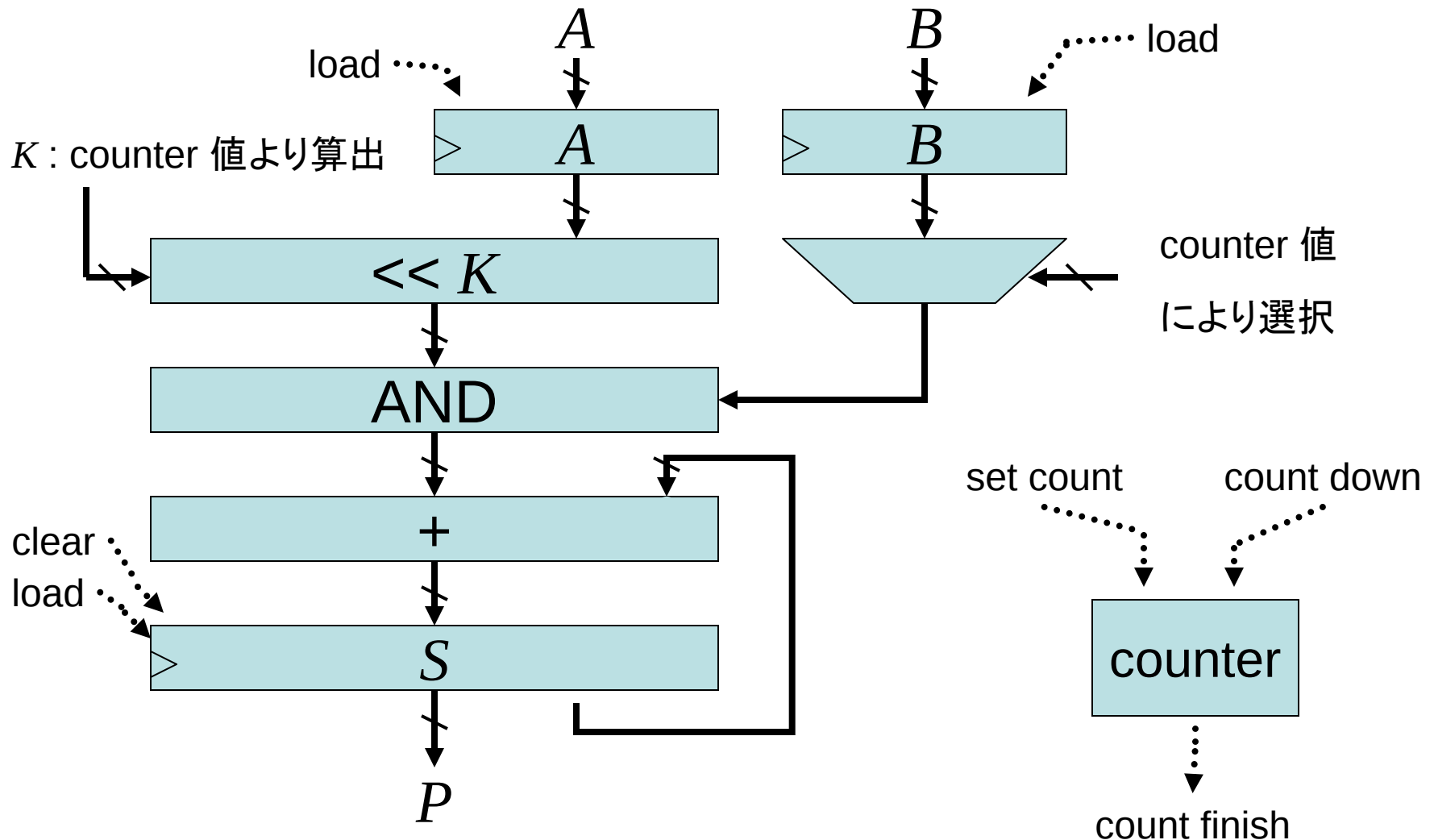
- 乗算の筆算の手順
 - 「シフト、論理積、加算」の繰り返し
- 1クロックあたり1段分の部分計算に分割
 - 1クロック分の段数はほぼ加算1段分
 - 段数削減 → 最大遅延減少 → クロック周波数の低下を防ぐ
 - しかし、全計算にかかる時間は増加する
 - 同じ計算の繰り返しなので、計算資源(シフタ、論理積演算器、加算器、レジスタ)を共有できる
 - 回路規模削減

※乗算器を持たない CPU で乗算を行なう場合の処理もこれと同じ

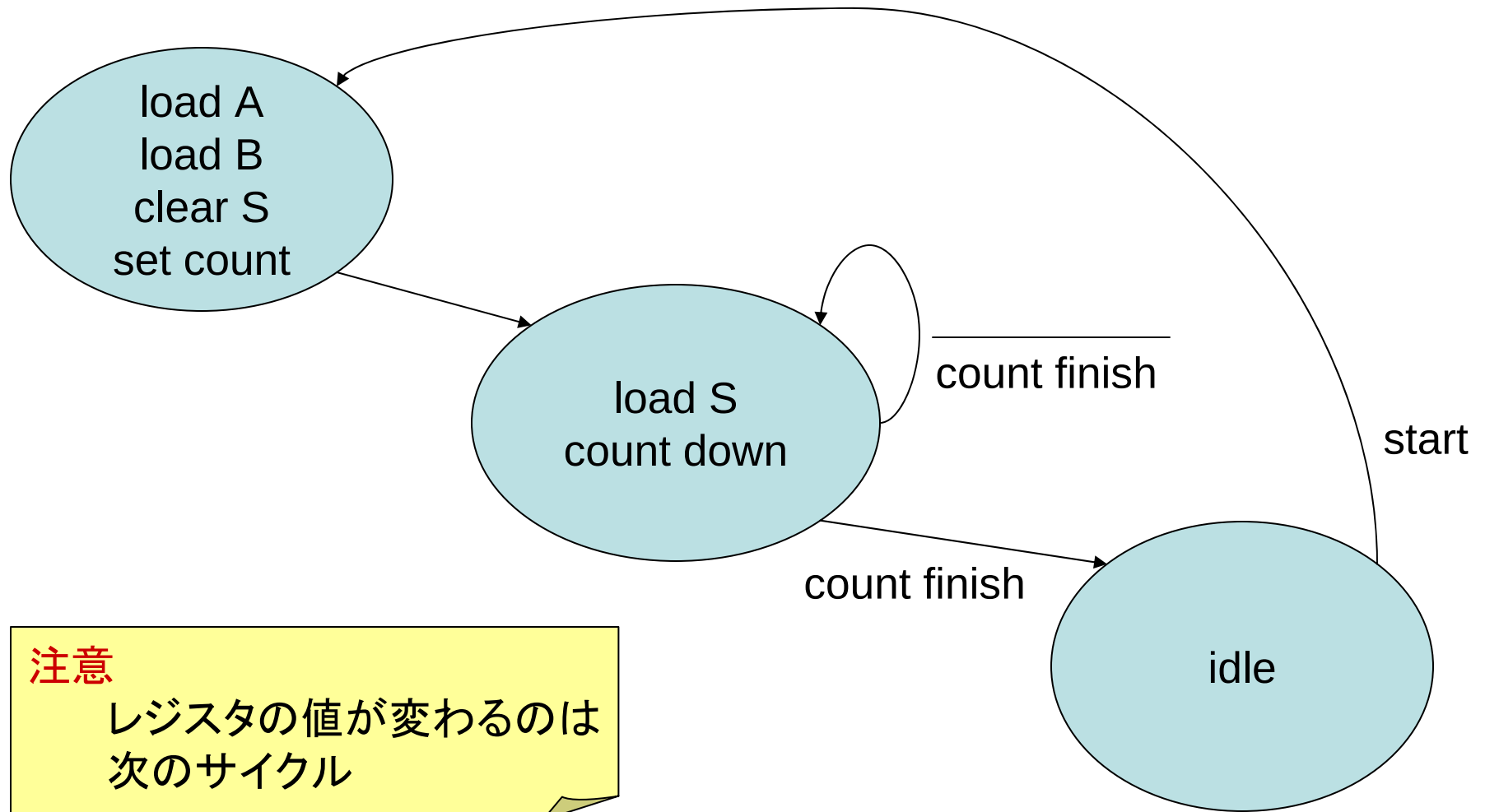
複数サイクル乗算器の単純構成 の要素例

- 被乗数 A を保持する n bit レジスタ
- 乗数 B を保持する n bit レジスタ
- 中間結果 S を保持する $2n$ bit レジスタ
- $2n$ bit バレルシフタ (シフト量 $0 \sim n - 1$)
- n bit セレクタ
- $2n$ bit AND
- $2n$ bit 加算器
- 現在何段目进行处理しているか保持するカウンタ ($\log n$ bit レジスタ、 $\log n$ bit 比較器)
- その他の制御回路 (状態遷移機械、制御ロジックなど)

複数サイクル乗算器の単純構成例



複数サイクル乗算器の状態遷移例



演習8A

- 4bit × 4bit の符号無し乗算を行なう単純構成の複数サイクル乗算器について、 1011×1101 を計算させたとき、それぞれのレジスタ値、信号値はいくらになるか、初期状態から結果が出てくるまでの毎サイクルについて答えよ。

演習8A(続き)

- 単純構成例について、 $n=4$ として
 - 図中の信号線はそれぞれ何ビットか、記入せよ。
 - セレクタ部分を基本ゲートを部品として設計せよ。
 - [AND] と描かれたモジュールを基本ゲートを部品として設計せよ。
 - カウンタ部分を、レジスタ、加減算器、セレクタ、基本ゲートを部品として設計せよ。
 - 制御回路を設計せよ。

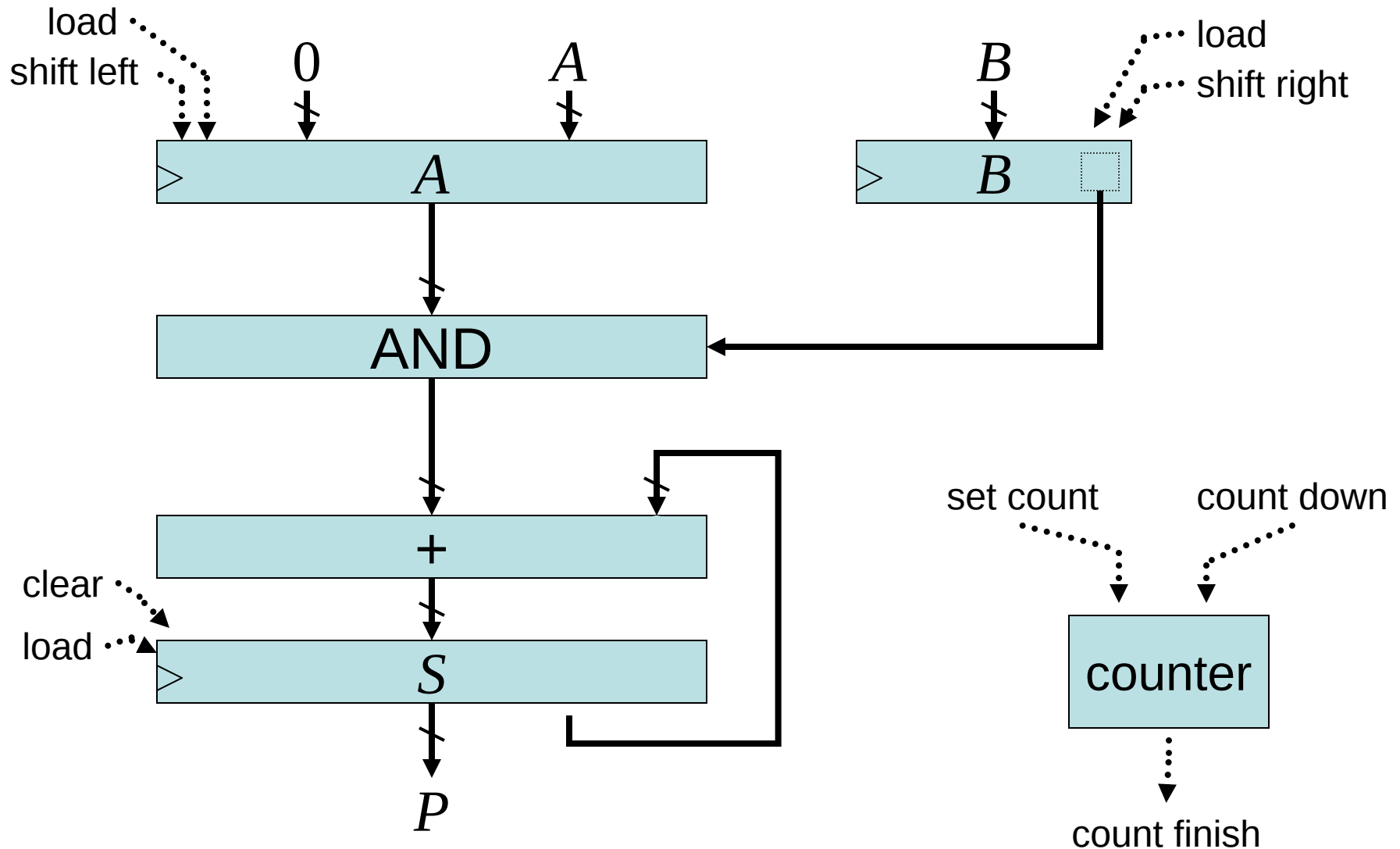
単純構成の問題点

- バレルシフタが必要
 - $2n$ bit セレクタ $\times \log n$ 段
- セレクタが必要
- シフトレジスタ化してみたら？

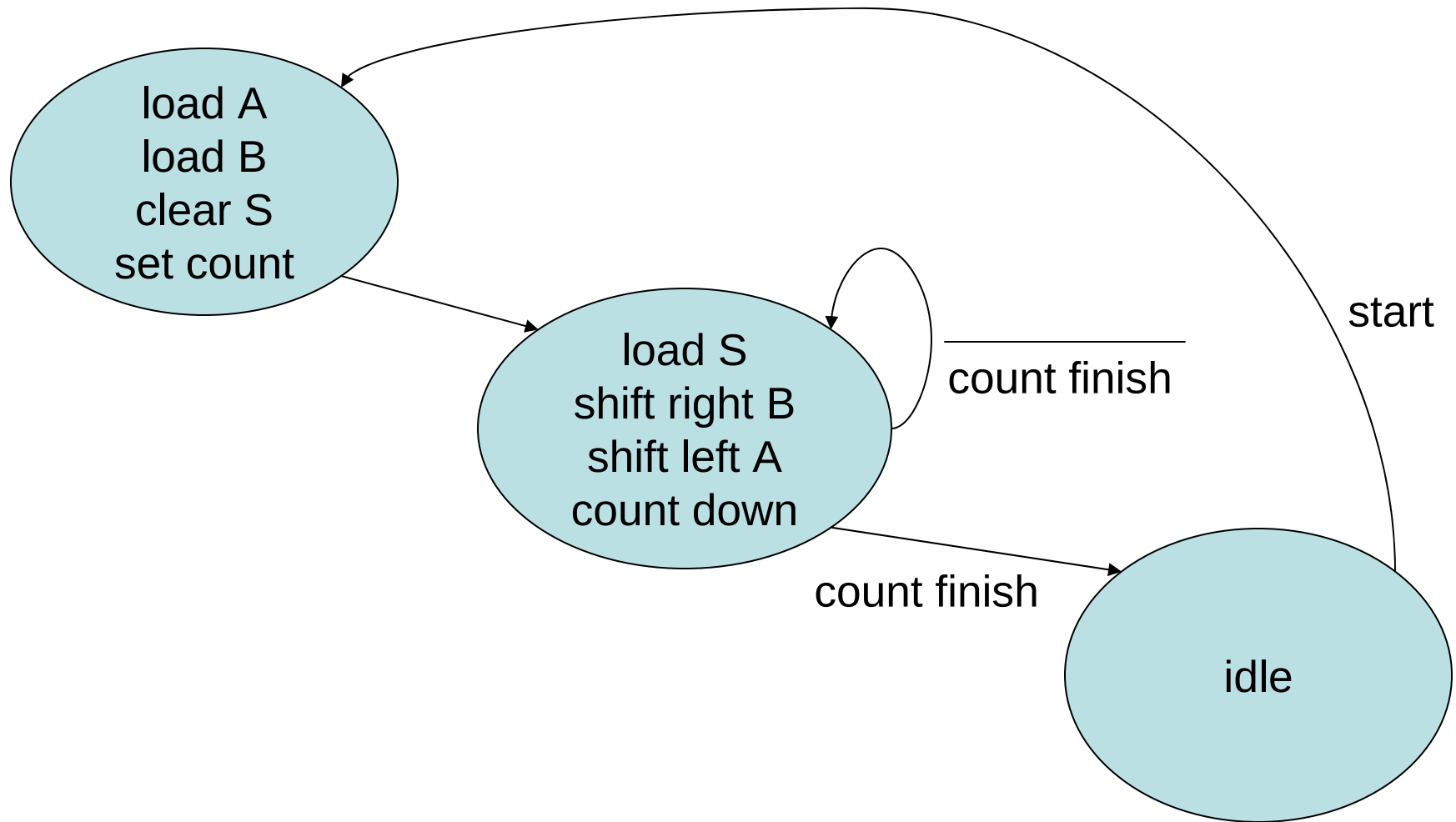
複数サイクル乗算器の要素例2

- 被乗数 A を保持する $2n$ bit シフトレジスタ
- 乗数 B を保持する n bit シフトレジスタ
- 中間結果 S を保持する $2n$ bit レジスタ
- $2n$ bit AND
- $2n$ bit 加算器
- 現在何段目进行处理しているか保持するカウンタ($\log n$ bit レジスタ、 $\log n$ bit 比較器)
- その他の制御回路(状態遷移機械、制御ロジックなど)

複数サイクル乗算器の構成例2



複数サイクル乗算器の状態遷移例2



演習8B

- 4bit × 4bit の符号無し乗算を行なう構成例2の複数サイクル乗算器について、 1011×1101 を計算させたとき、それぞれのレジスタ値、信号値はいくらになるか、初期状態から結果が出てくるまでの毎サイクルについて答えよ。

演習8B(続き)

- 構成例2について、 $n=4$ として
 - 図中の信号線はそれぞれ何ビットか、記入せよ。
 - 制御回路を設計せよ。

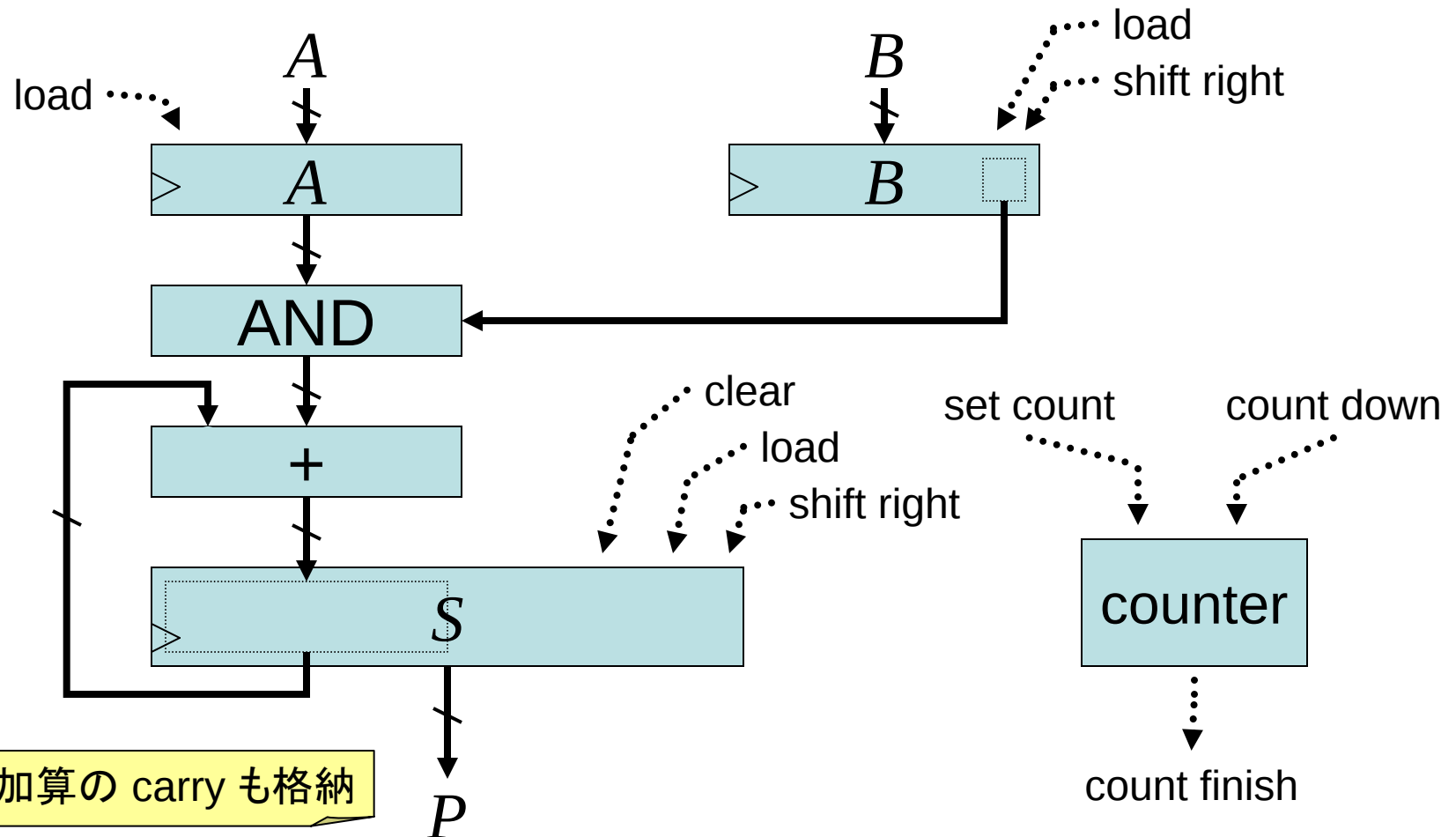
構成例2の問題点

- 被乗数 A を格納するレジスタ、AND、加算が無駄に大きい。
 - 実質的には n bit 分の計算なのに $2n$ bit 分
- 逆に、合計を保存するレジスタ S の方をシフトしてみたら？

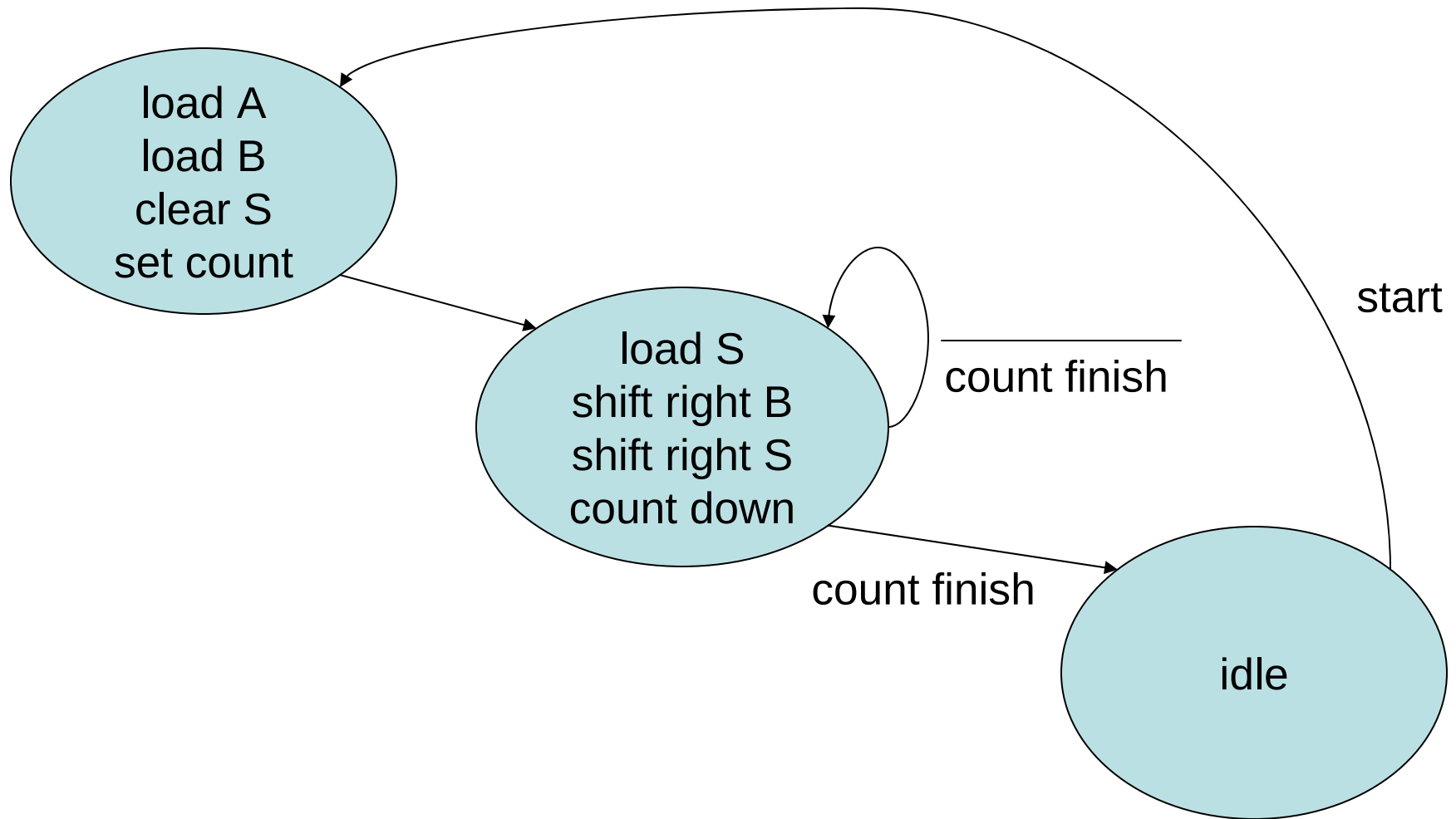
複数サイクル乗算器の要素例

- 被乗数 A を保持する n bit レジスタ
- 乗数 B を保持する n bit シフトレジスタ
- 中間結果 S を保持する $2n$ bit シフトレジスタ
- n bit AND
- n bit 加算器
- 現在何段目进行处理しているか保持するカウンタ ($\log n$ bit レジスタ、 $\log n$ bit 比較器)
- その他の制御回路 (状態遷移機械、制御ロジックなど)

複数サイクル乗算器の構成例



複数サイクル乗算器の状態遷移例



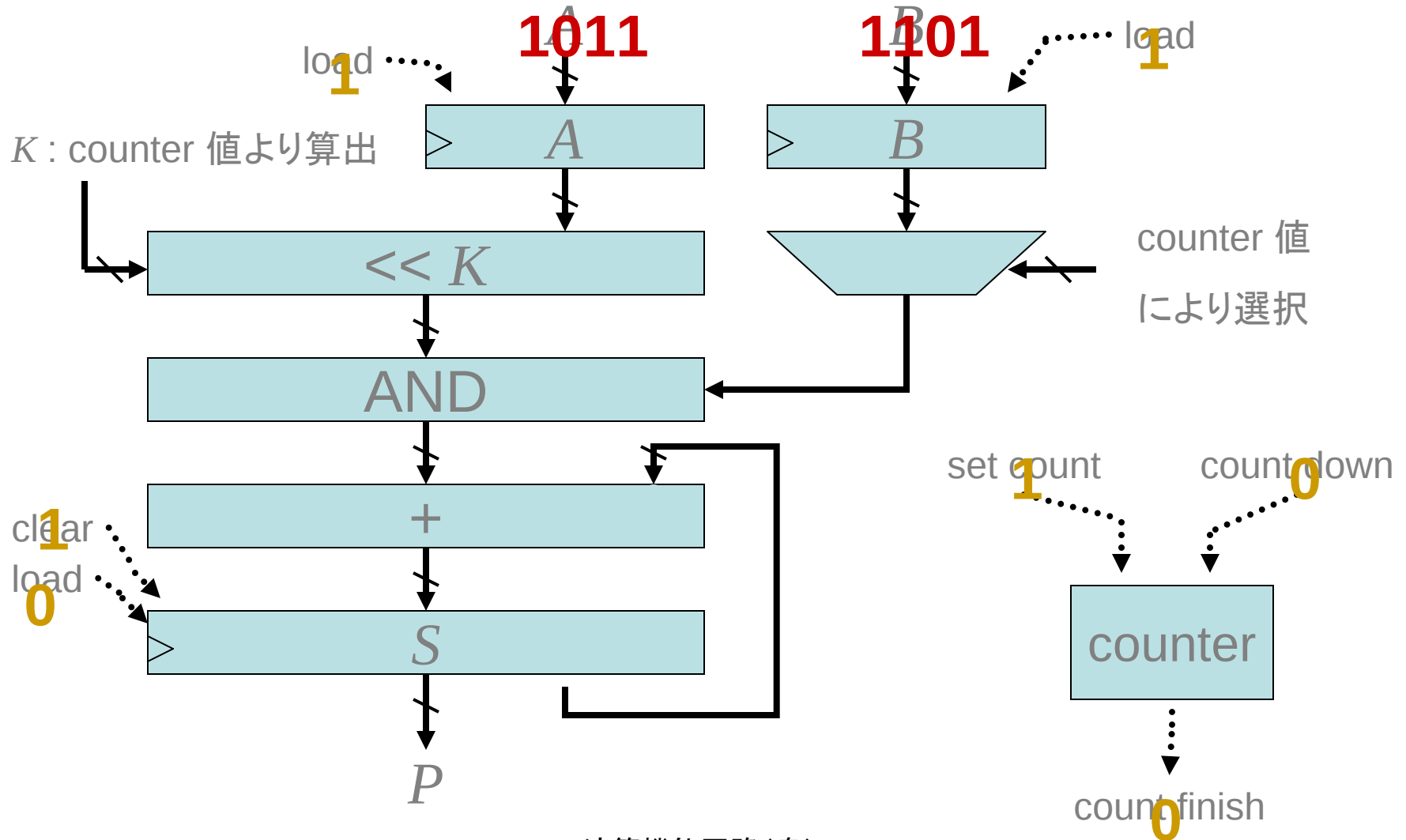
演習8C

- 4bit × 4bit の符号無し乗算を行なう最終構成例の複数サイクル乗算器について、 1011×1101 を計算させたとき、それぞれのレジスタ値、信号値はいくらになるか、初期状態から結果が出てくるまでの毎サイクルについて答えよ。

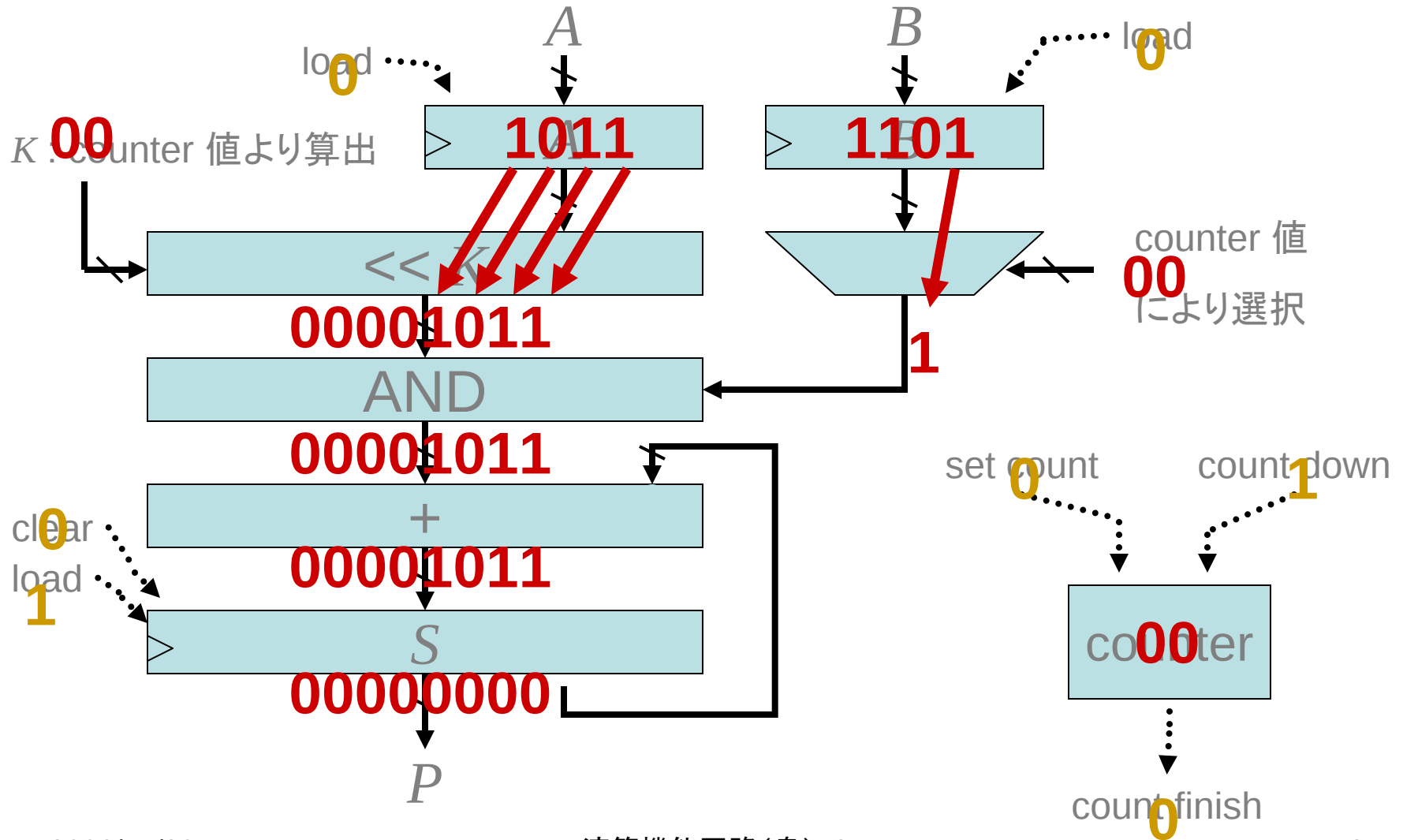
演習8C(続き)

- 最終構成例について、 $n=4$ として
 - 図中の信号線はそれぞれ何ビットか、記入せよ。
 - 制御回路を設計せよ。

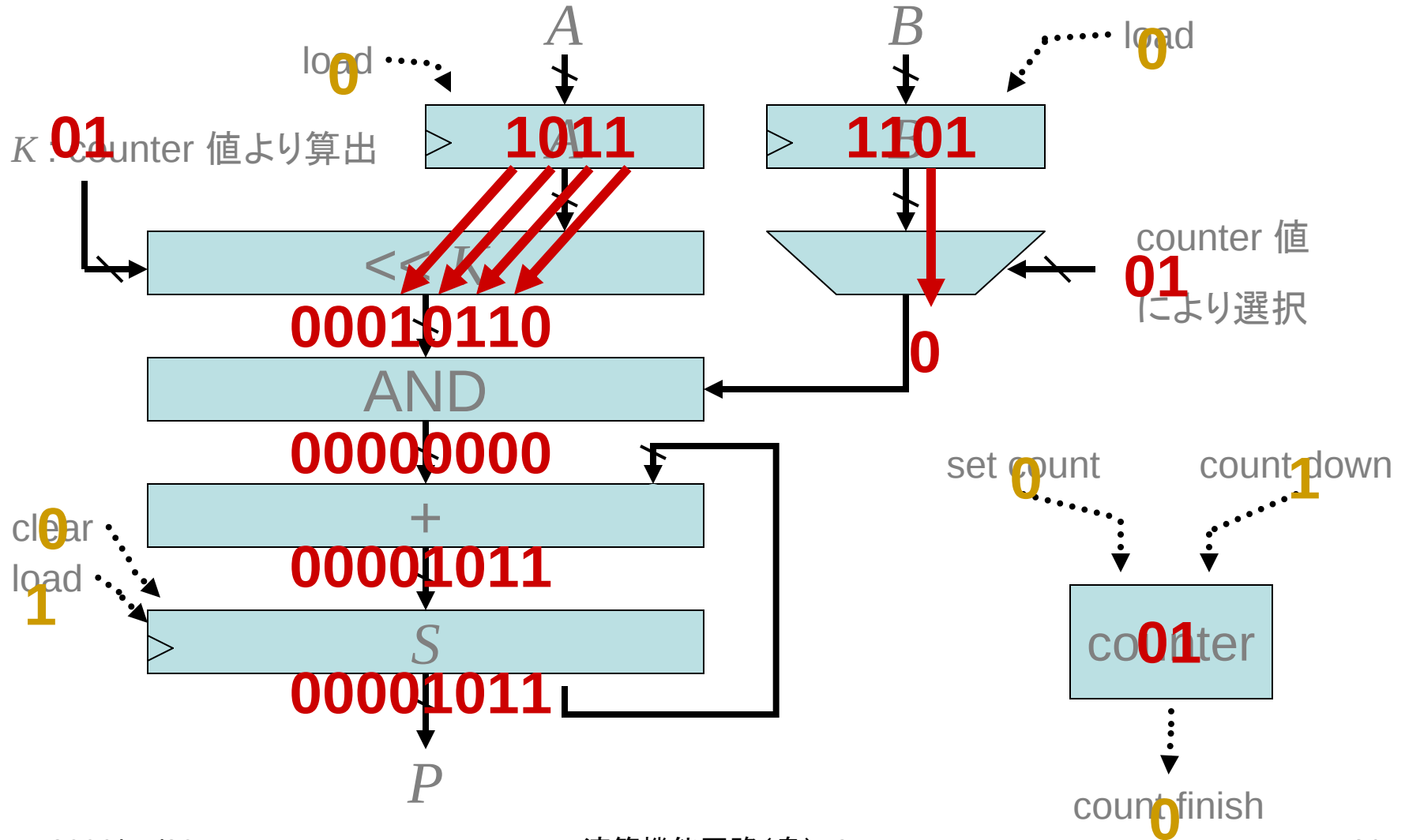
演習8A回答例(第1クロック中)



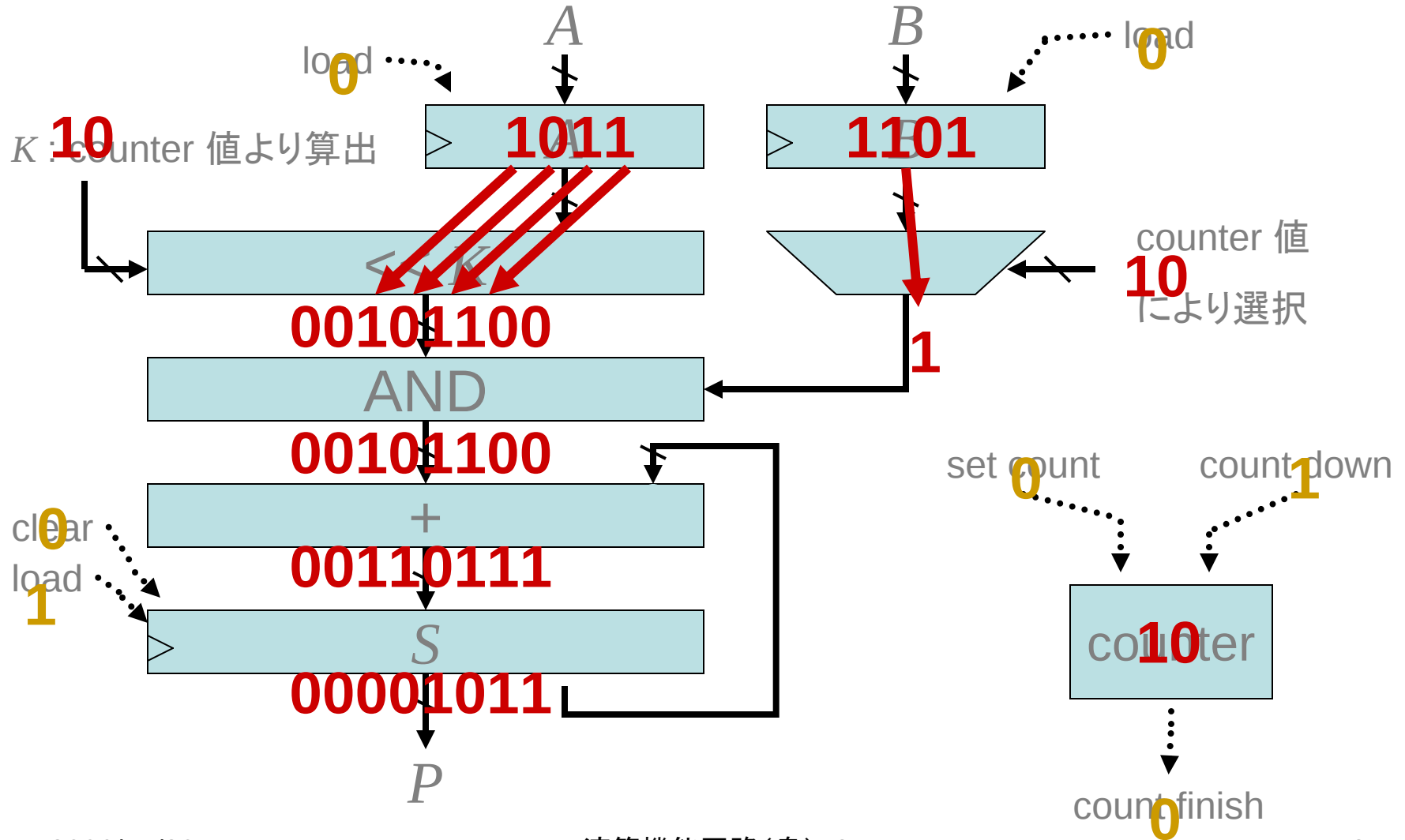
演習8A回答例(第2クロック中)



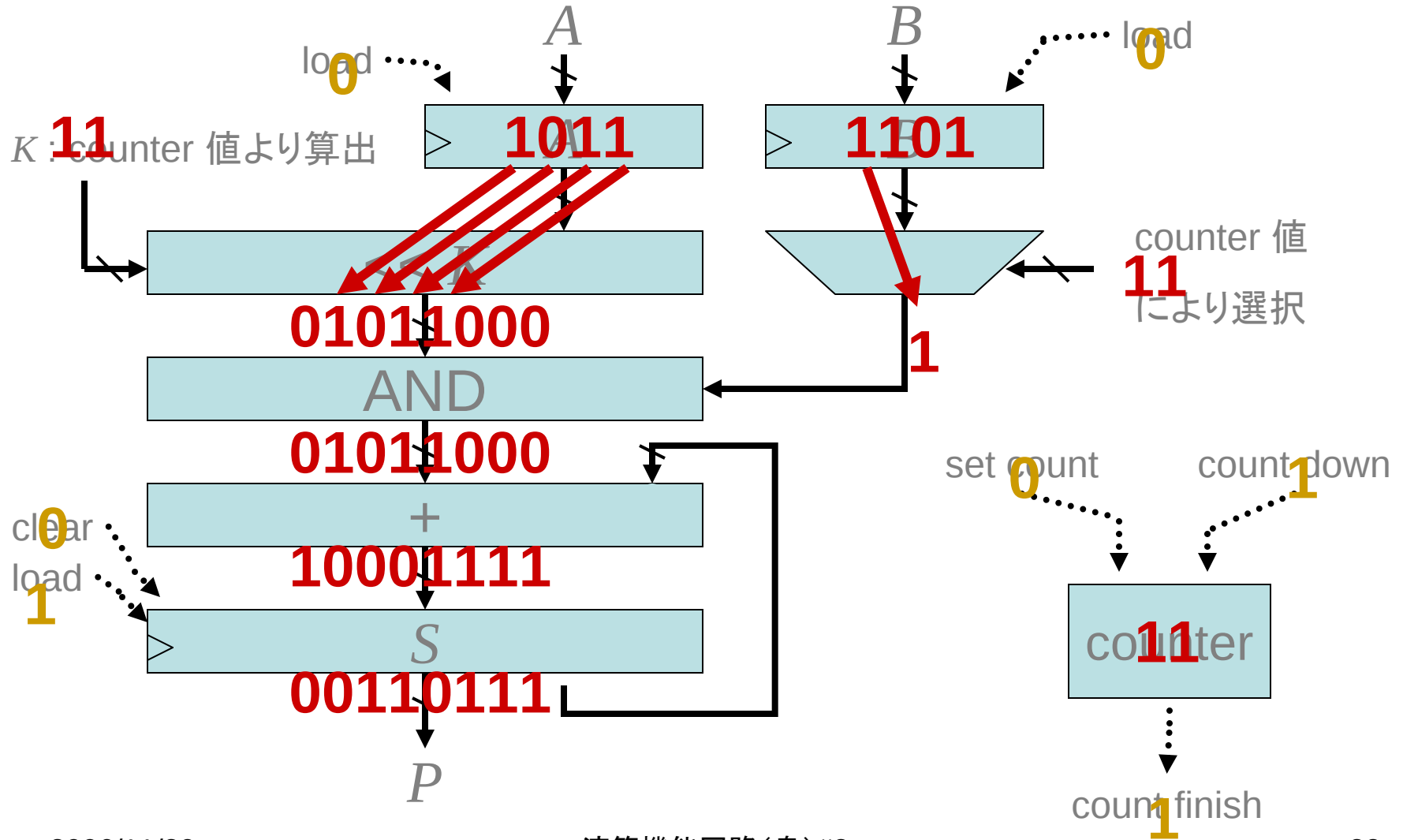
演習8A回答例(第3クロック中)



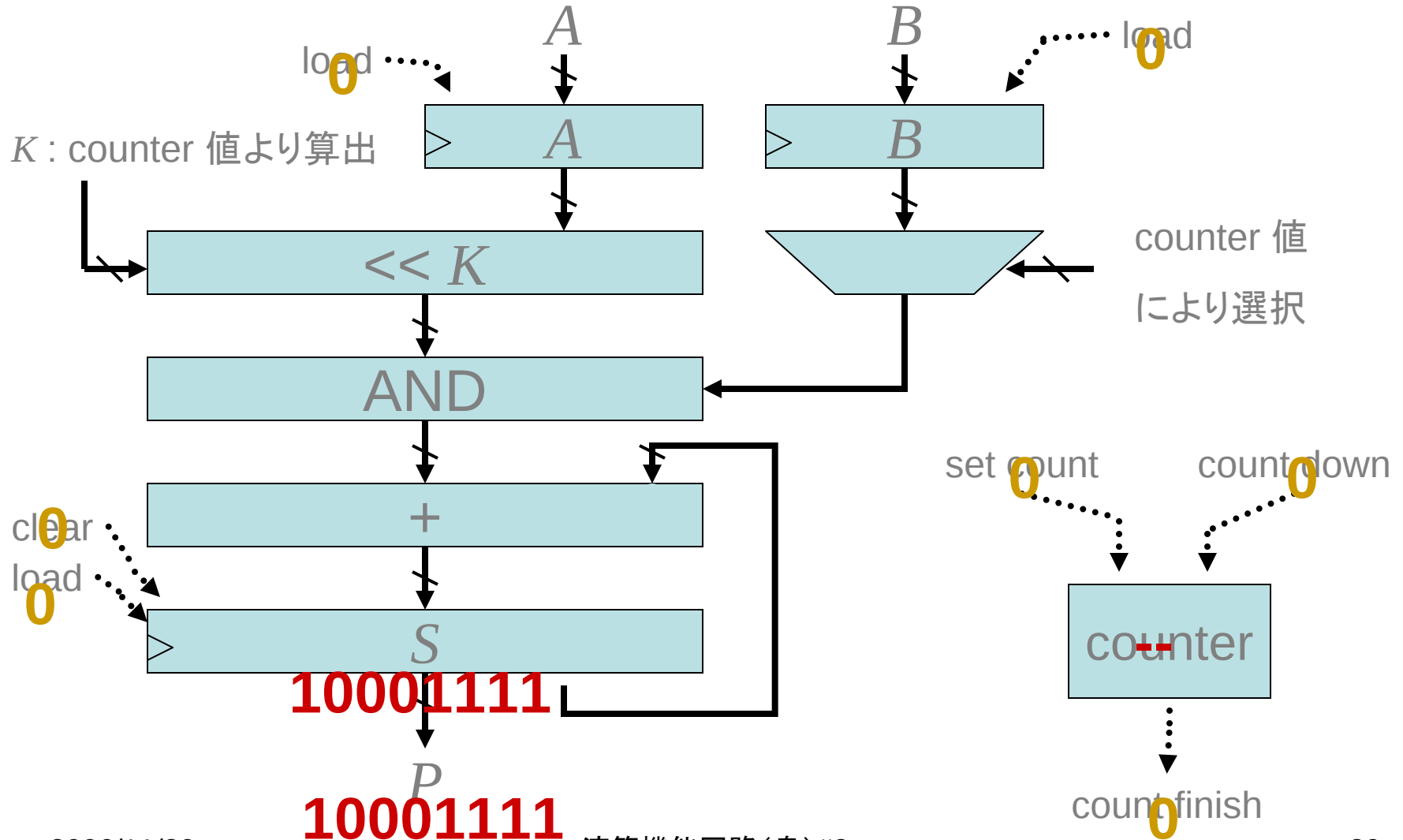
演習8A回答例(第4クロック中)



演習8A回答例(第5クロック中)



演習8A回答例(第6クロック中)



複数サイクル乗算器に関する補足

- 1サイクルあたり1段分では時間的余裕を取り過ぎ？
 - 1サイクルあたり複数段に改良することも可能
 - ブースの算法 (Booth's algorithm) などの技術を利用できる(必要に応じて各自調査されたし)
- Wallace Tree は向かない
 - ←各段の計算の数や組合せが異なる
 - ←保持すべき中間結果が多い