演算機能回路

第9週

2006/11/27

泉知論

立命館大学 理工学部 電子情報デザイン学科

パイプライン演算

pipelined operation

複数サイクル vs パイプライン

- 複数サイクル処理
  - 現在の計算が完了するまで、次の計算を投入できない
- 現在の計算が完了していなくても、次の計算を投入できたら、性能が上がる！

→パイプライン処理

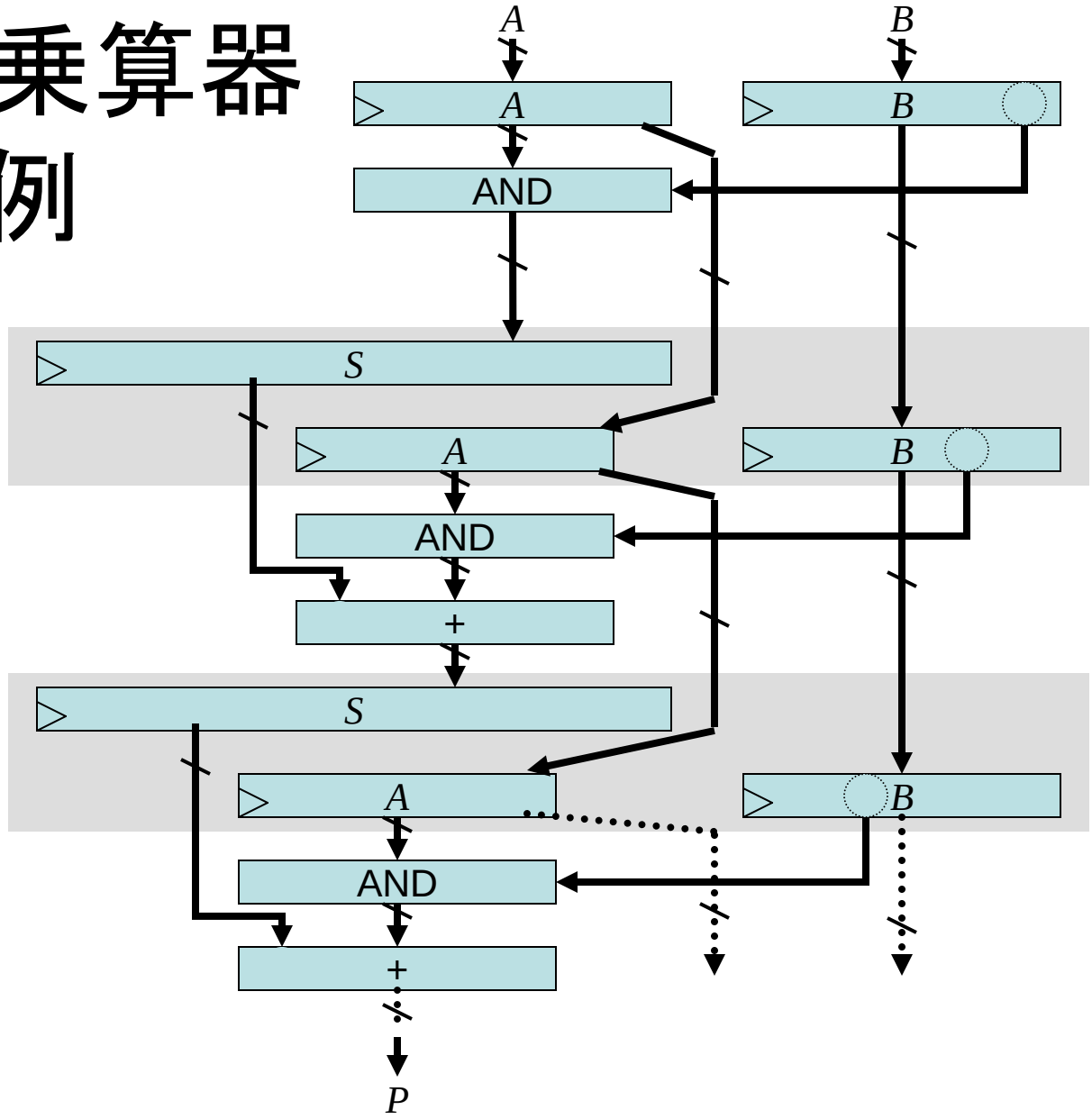
パイプライン演算

- 部分計算への分割
  - それぞれの部分計算を実行する回路(パイプラインステージ)を用意する
  - 中間値をレジスタに保持
  - 複数サイクルで計算
  - (ステージがあいたら)以前の計算の完了をまたずに次の計算を投入できる
  - 計算の投入回数増加→高性能化
  - 各ステージを回路化→回路規模増大
  - ひとつの計算の全てにかかる時間→減らない

パイプライン乗算器の構成要素例

- 被乗数 A を保持する n bit レジスタ
  × n ステージ分
- 乗数 B を保持する n bit レジスタ
  × n ステージ分
- 中間結果 S を保持する $2n$ bit レジスタ
  × n ステージ分
- n bit 加算器
  × n ステージ分
- その他制御回路など

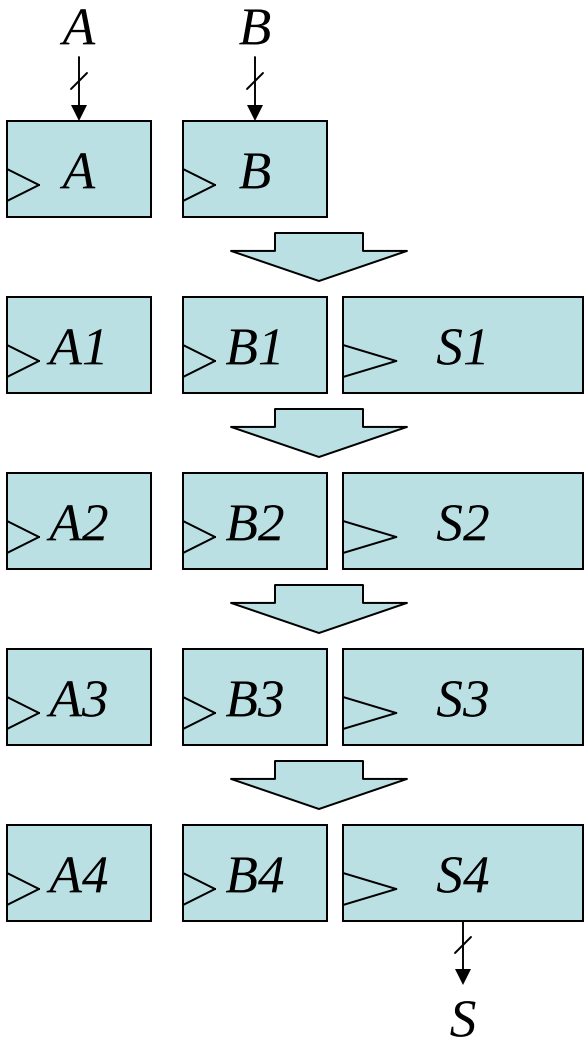
パイプライン乗算器の構成例



演習9A

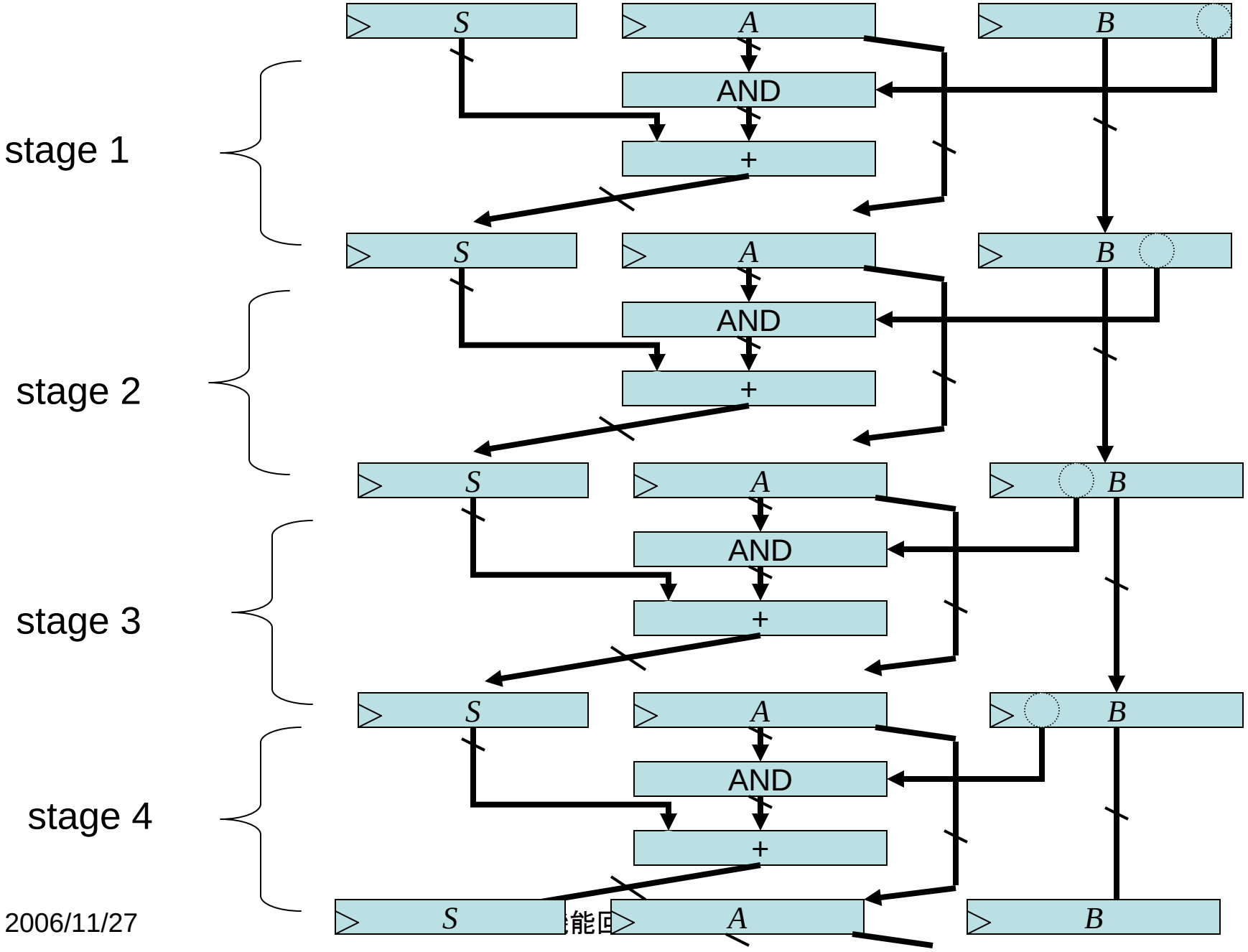
- 4bit × 4bit の符号無し乗算を行なうパイプライン乗算器について、１×２、３×４、５×６、…を順次投入していったときの各段のパイプラインレジスタの値はいくらになるか、初期状態（すべてとする）から、３×４の結果が出てくるまでの毎サイクルについて答えよ。

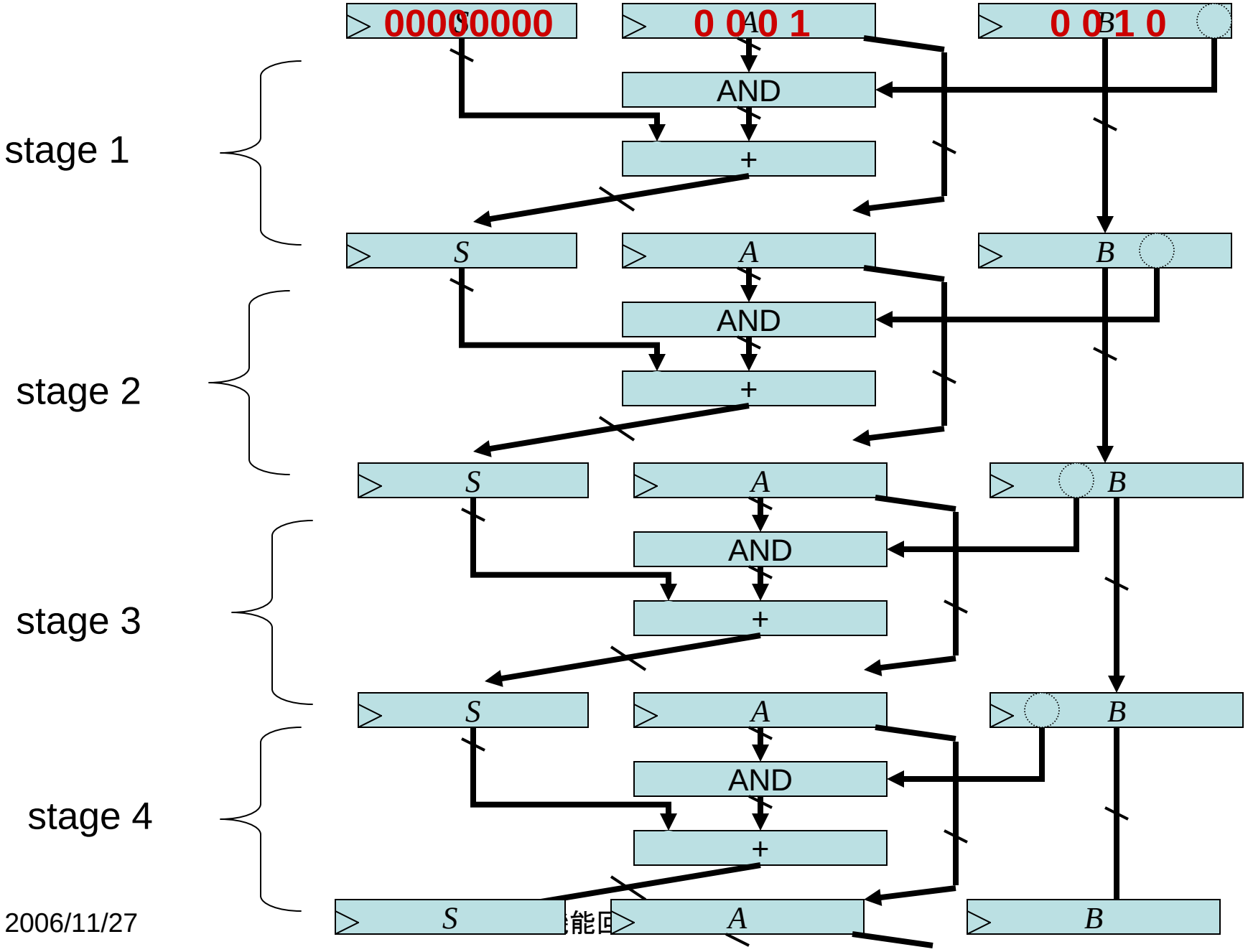
※回路構成上、工夫の余地はあるが、単純な構成でよい。

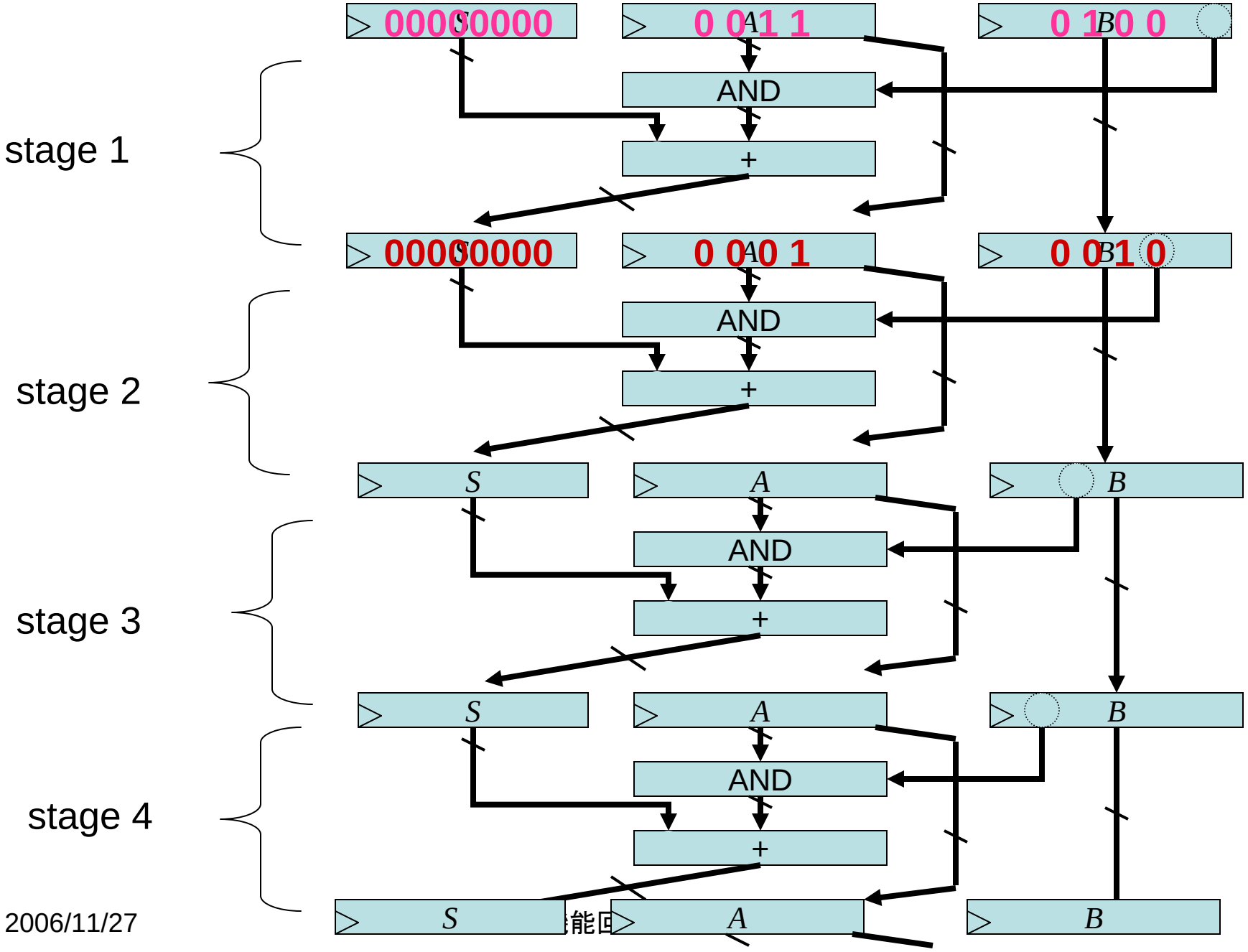


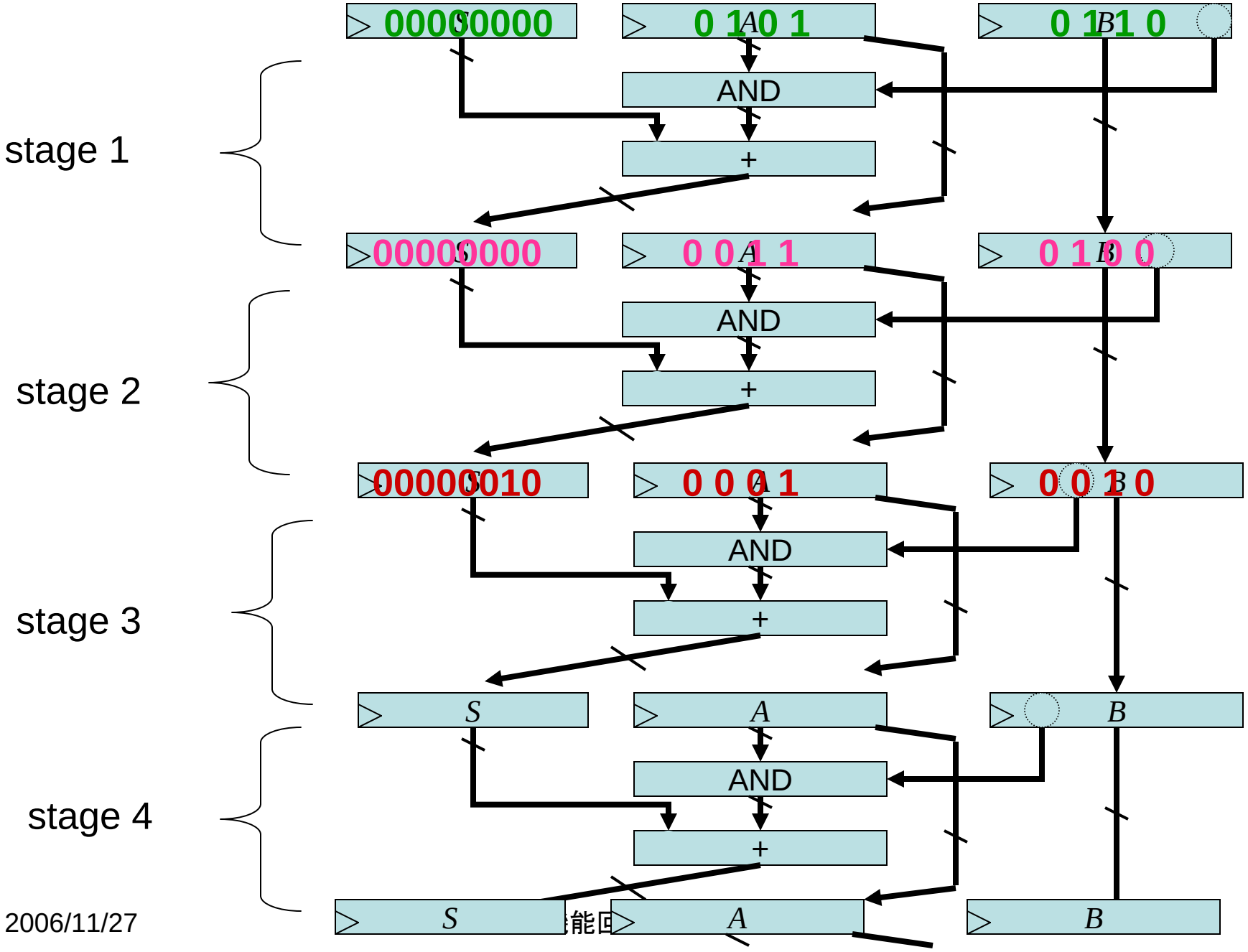
演習9A（続き）

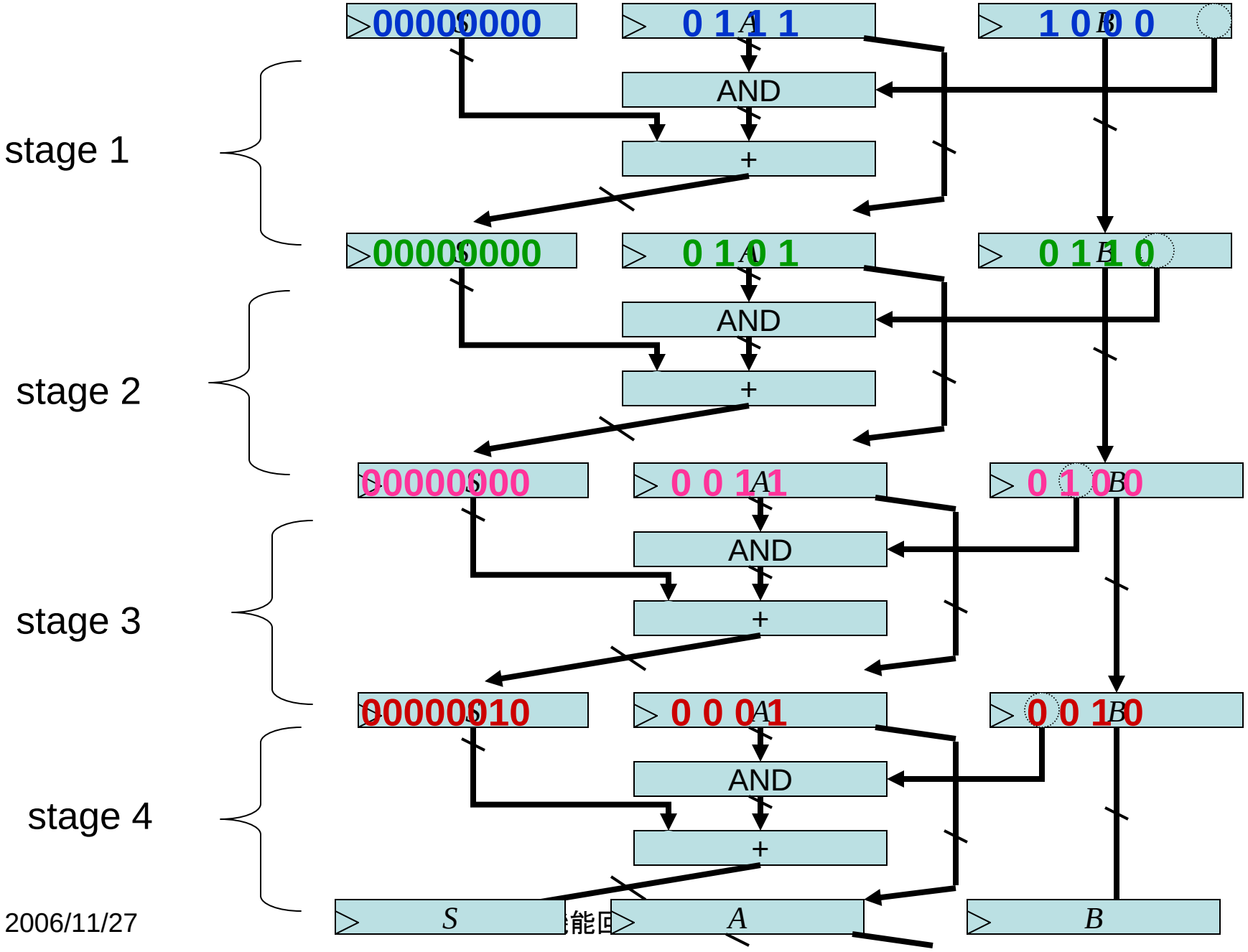
- 4 bit × 4 bit の符号無しパイプライン乗算器を、レジスタ、加減算器、セレクタ、基本ゲートを部品として用いて設計せよ。回路構成上、工夫の余地はあるが、単純な構成でよい。
- 回路規模削減、速度（レイテンシ）削減のための工夫について検討せよ。


stage 1
stage 2
stage 3
stage 4
S
A
B
AND
+


stage 1
stage 2
stage 3
stage 4
00000000
0 0 0 1
0 0 1 0
AND
+
S
A
B


stage 1
stage 2
stage 3
stage 4
S
A
B
AND
+
0000 0000
0 0 1 1
0 1 0 0
0000 0000
0 0 0 1
0 0 1 0


stage 1
stage 2
stage 3
stage 4
00000000
S
0 1 0 1
A
0 1 1 0
B
AND
+
00000000
S
0 0 1 1
A
0 1 0 0
B
AND
+
00000010
S
0 0 0 1
A
0 0 1 0
B
AND
+
S
A
B
AND
+
S
A
B


stage 1
stage 2
stage 3
stage 4
S
A
B
AND
+
00000000
0 1 1 1
1 0 0 0
00000000
0 1 0 1
0 1 1 0
00000000
0 0 1 1
0 1 0 0
00000010
0 0 0 1
0 0 1 0
S
A
B

パイプライン乗算器に関する補足

- 1ステージあたり1段分では時間的余裕を取り過ぎ？
  - 1ステージあたり複数段に改良可能
  - ブースの算法 (Booth’s algorithm) など
- 複数サイクル化との複合も可能
  - 各ステージを複数サイクル演算とすることで
- Wallace Tree は通常用いない

  ←保持すべき中間結果が多い