

電気電子物理工学実験Ⅲ

マイクロプロセッサ

課題1: 8ビット乗算高速化

マイクロプロセッサ(デジタル論理回路)で乗算を実行 ⇒ 筆算と同じ方式で行う

4ビット数 × 4ビット数の例

部分積を加算

				1	1	0	1	=13
×				1	0	1	1	=11
				1	1	0	1	=13 × 2 ⁰ × 1 ①
			1	1	0	1	0	=13 × 2 ¹ × 1 ②
		0	0	0	0	0	0	=13 × 2 ² × 0 ③
	1	1	0	1	0	0	0	=13 × 2 ³ × 1 ④
1	0	0	0	1	1	1	1	=143

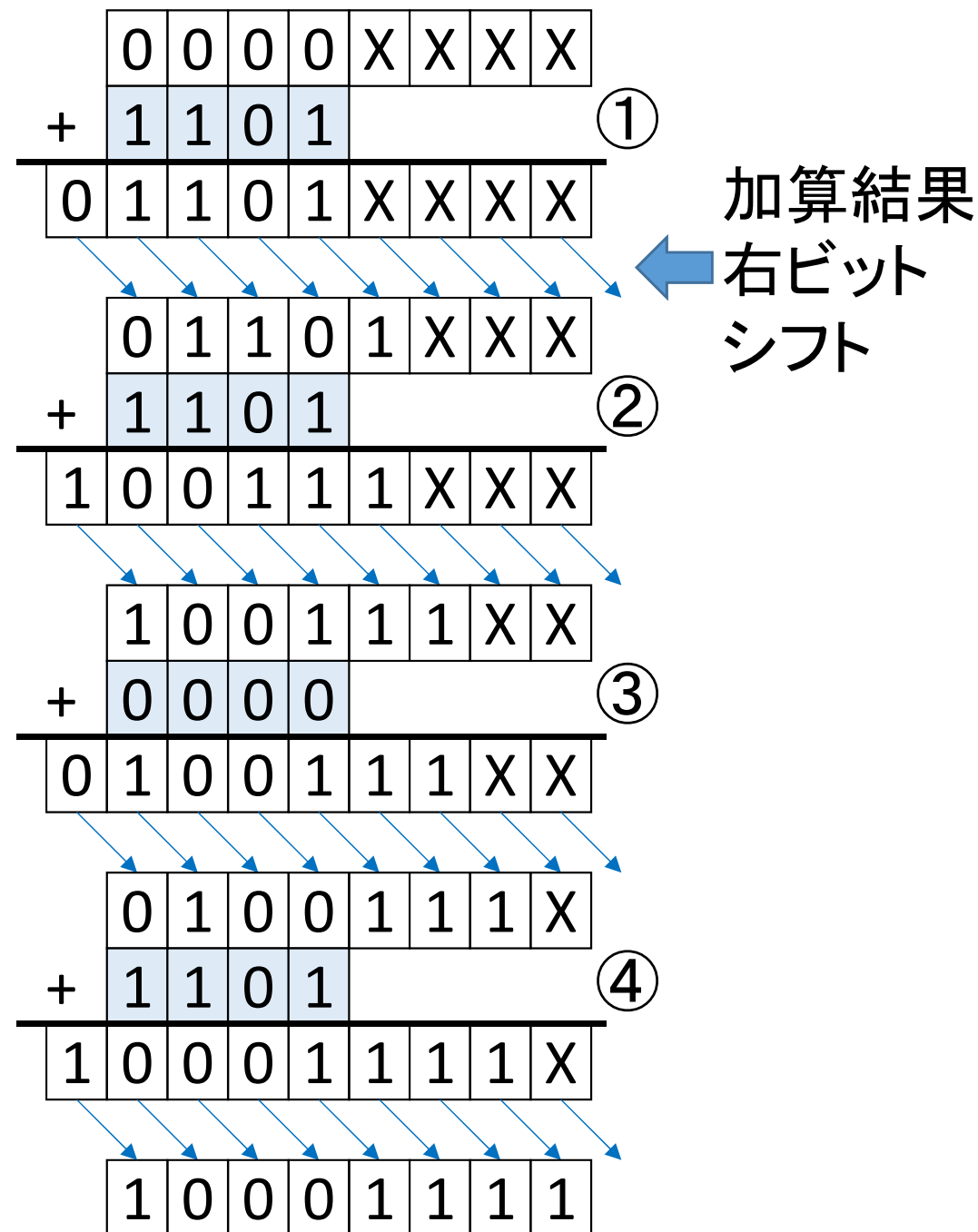
					0	0	0	0				
+						1	1	0	1			
					0	0	0	0	1	1	0	1

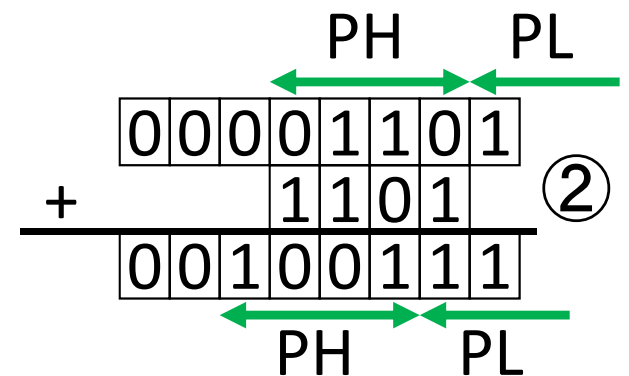
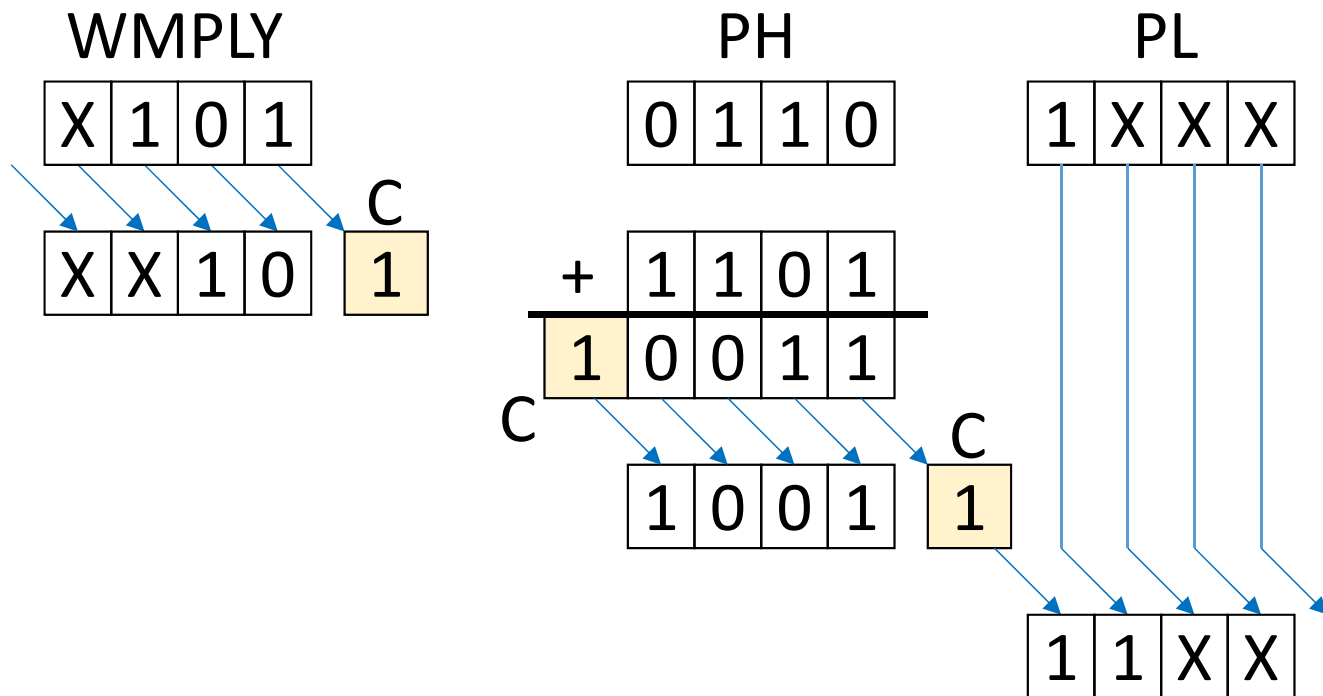
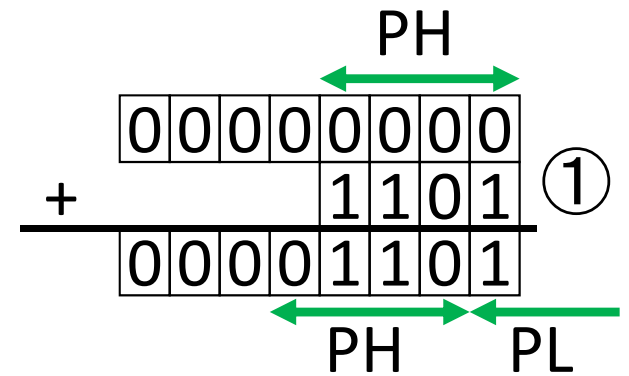
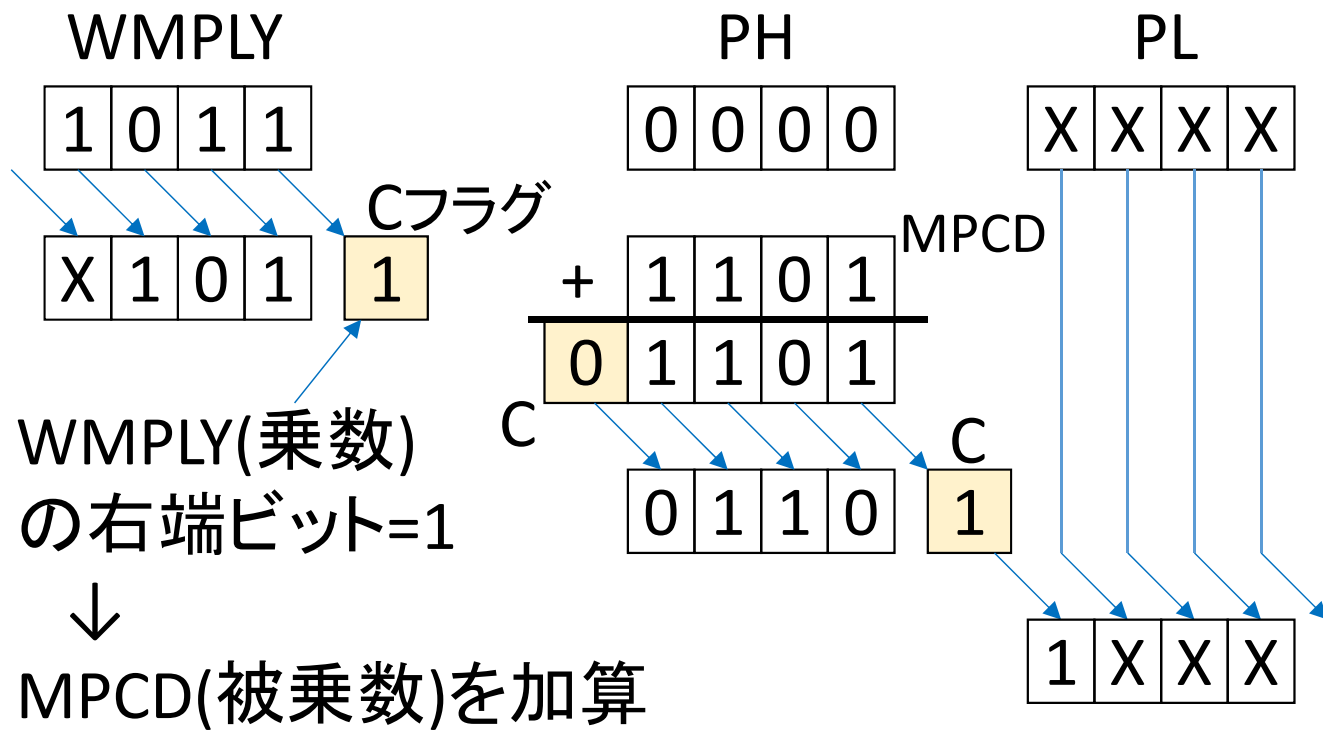
					0	0	0	0	1	1	0	1
+							1	1	0	1		
					0	0	1	0	0	1	1	1

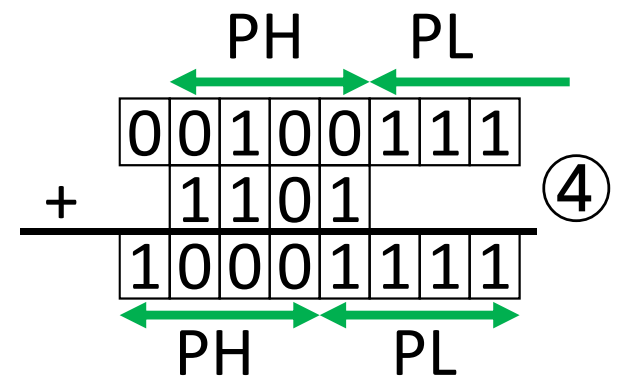
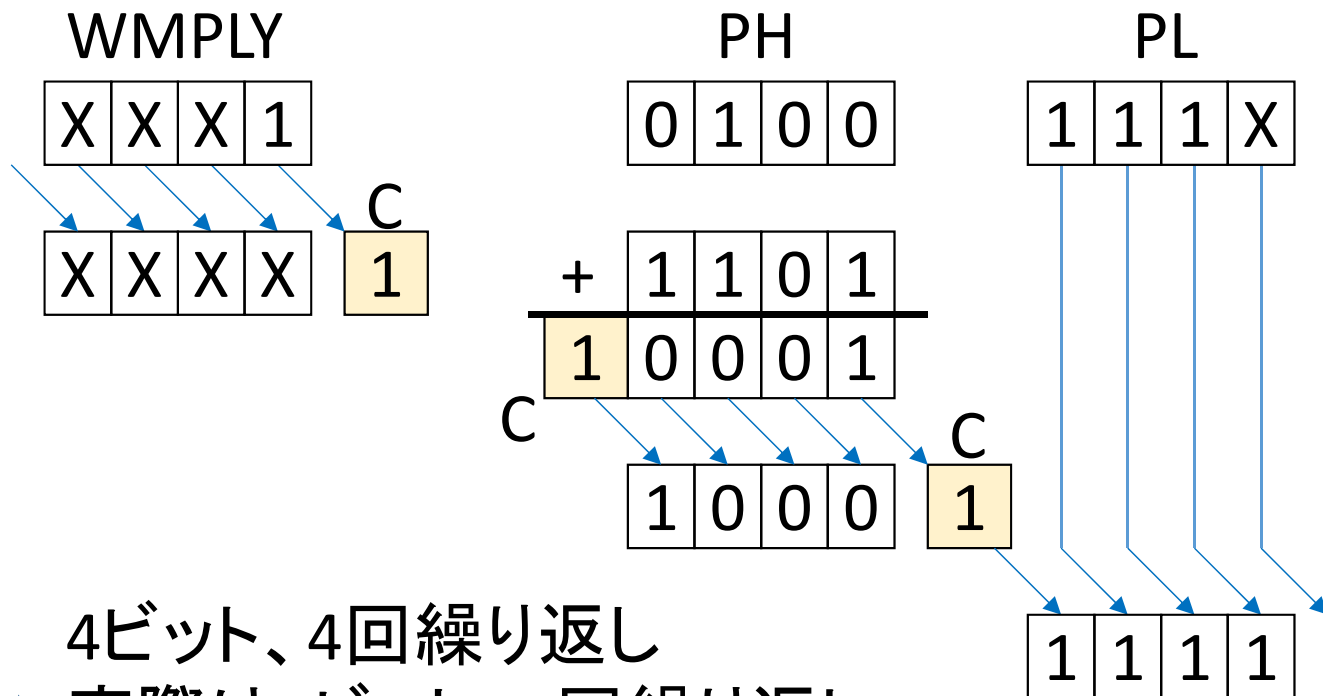
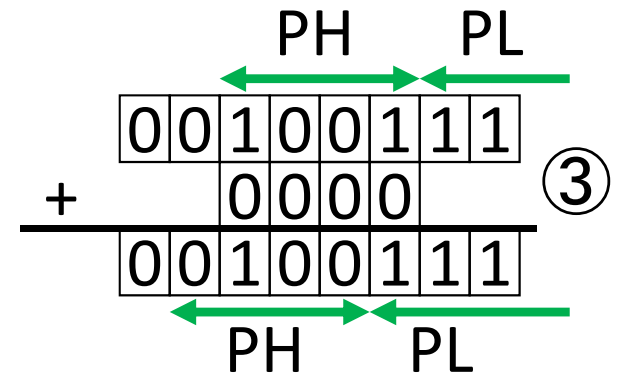
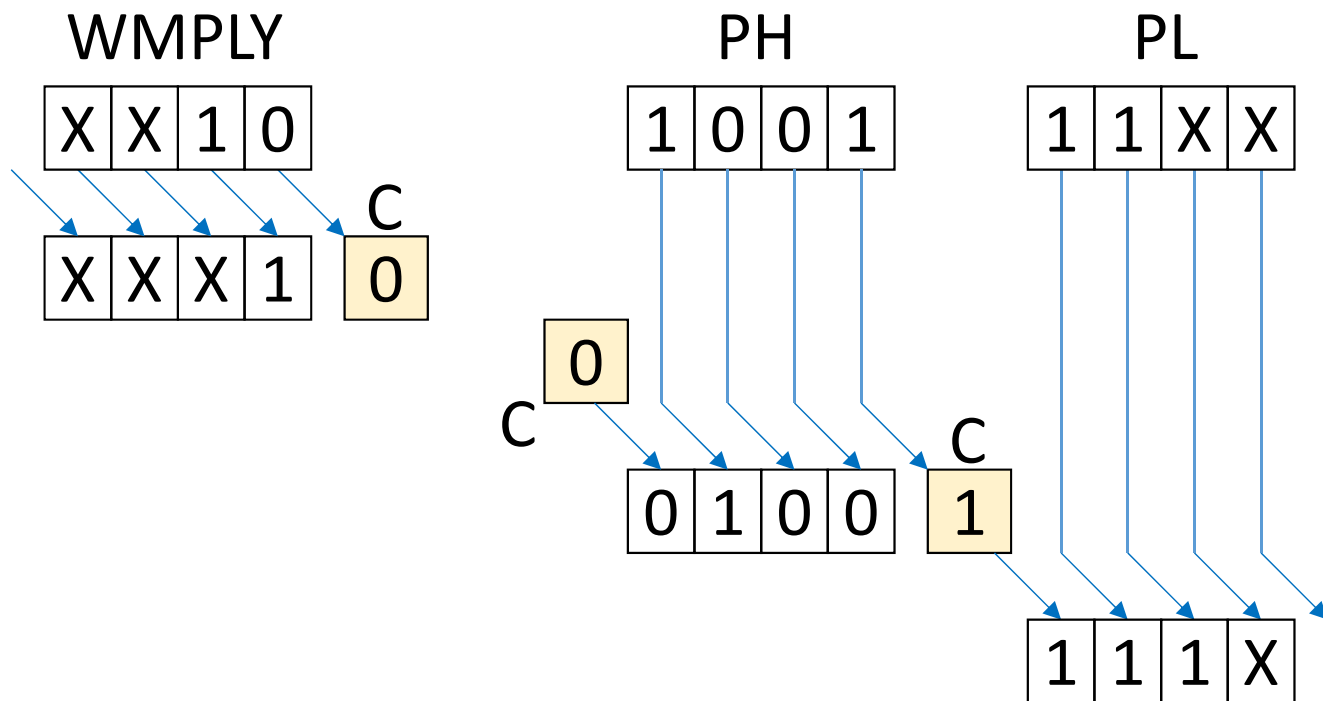
					0	0	1	0	0	1	1	1
+							0	0	0	0		
					0	0	1	0	0	1	1	1

					0	0	1	0	0	1	1	1
+						1	1	0	1			
					1	0	0	0	1	1	1	1

筆算では部分積を順に左に
ビットシフトして加算するが、
逆に加算結果を右にビット
シフトして加算することもできる







4ビット、4回繰り返し
 ➡ 実際は8ビット、8回繰り返し

クロックサイクル数を削減するには

• レジスタを活用しメモリアクセスを減らす

- メモリ (WMPLY,PH,PL,COUNT)の読み書きアクセスはメモリアドレスの準備などにクロックサイクルを消費
- レジスタもデータ記録領域であり、メモリよりも簡単に読み書き可能なためクロックサイクルが削減
- サンプルプログラムではB,WA,Xレジスタが未使用

• 16ビット処理を利用

- 8ビット数乗算結果は16ビット数になるが、サンプルプログラムでは8ビットずつ処理しており、手間をかけている
- ELC-1は16ビット処理が可能であり、16ビット処理を活用してクロックサイクルを削減

• より複雑な処理を行う命令を定義・利用

- 機械語命令はフェッチとデコードにクロックサイクルを消費
- 複数命令の処理内容を1つの命令にまとめることでクロックサイクルを削減

その他

- 何が何だかさっぱり分からないときは
 - これまで講義等で聞いたことのない内容だと思うので、そう思うのも当然です。
 - 個別に分かるまで説明しますので申し出てください。
- プログラム実行結果がおかしいときは
 - シミュレータを再起動してみましょう。
 - 自分のプログラムをよく確認しましょう。
 - それでも納得できないときは、シミュレータのバグが原因の可能性もあります。
 - 「こういう時にこの命令を実行すると期待と違う結果になる」というように、具体的に何がどうおかしいのか示してください。