

マイクロコンピュータとアセンブリ言語 プログラム実習課題用紙（2週目用）

提出者	学籍番号		氏名		机番号	
共同作業	学籍番号		氏名			

次の①から④までを繰り返し実行するプログラム 8ビット2進加算プログラム (rei3.a30) を改造して、下記各設問の①から④まで連続動作するようなプログラムを作成せよ。

rei3.a30 の動作解説:

- ① 400H 番地の初期状態は 0H とする。SW 9 (port8 の bit2) を押下した時、トグル SW (port3) で設定した数値を 400H 番地に保存する。このとき SW を ON したとき (LCD 方向に倒したとき) を 1 とする。続いて②へ。
- ② ①で保存したデータを LED 1 ~ LED 8 に表示する。③へ。
- ③ SW 10 (port8 の bit3) を押下した時、トグル SW (port3) で設定した数値を読む。④へ。
- ④ ①で保存した値と③で読んだ数の加算を行い、①へ戻る

i) 8ビット減算(④のヒント: テキスト p33 表 8 で適切な命令を探して rei3.a30 を変更)

- ①から③まで同様。④へ続く。
- ④ ①で保存した数値と③で読み込んだ数値を使って 8ビット同士の**減算**を行い、結果を LED 1 ~ LED 8 に表示する。①で保存した値を被減数、③の演算数を減数とする。

ii) 8ビット左論理シフト(ヒント: p15 ビットマスク操作で下位 3bit だけ取り出すには?)

(さらにヒント: 左論理シフトを行うための命令には、対象とするレジスタに制限がある)

- ①から③まで同様。④へ続く。
- ④ ①で保存した数値と③で読み込んだ数値を使って 8ビットの**左論理シフト**を行い、結果を LED 1 ~ LED 8 に表示する。①で保存した値を、③で読み込んだ値の数だけシフトさせる。③で読み込んだ値は 1 ~ 7 まで(下位 3ビット分だけ)にマスクすること。

動作確認の例: 一回目の入力 0000 0101 ⇒ここで sw9 押し
 二回目の入力 1000 0100 (※下位 bit で「4」を指定) ⇒ここで sw10 押し
 結果の表示 0101 0000 (※「4」bit だけ左にずれた)

iii) 8ビット右算術シフト(ヒント: 右シフトするには? マイナスの数を表現するには?)

- ①から③まで同様。④へ続く。
- ④ ①で保存した数値と③で読み込んだ数値を使って 8ビットの**右算術シフト**を行い、結果を LED 1 ~ LED 8 に表示する。①で保存した値を、③で読み込んだ値の数だけシフトさせる。③で読み込んだ値は 1 ~ 7 まで(下位 3ビット分だけ)にマスクすること。

動作確認の例: 一回目の入力 1011 1000 (※先頭が 1 なのでマイナスの数)⇒ここで sw9 押し
 二回目の入力 1000 0011 (※下位 bit で「3」を指定) ⇒ここで sw10 押し
 結果の表示 1111 0111 (※「3」bit だけ右にずれた)

(応用課題 1) (ヒント: 条件ジャンプ命令を使う)

iii) のプログラムで、演算を行った結果のキャリーフラグの状態を CPU ボード上の LED (port0 の bit0) へ表示する機能を追加せよ。

(応用課題 2) (ヒント: テキスト p17, 繰り返し処理の仕方は先週の問題 5. iii) を参考に)

トグルスイッチ (port3) で設定した 2進数を SW 9 (port8 の bit2) を押下した時に、BCD 変換し、結果を LED 1 ~ LED 8 に表示せよ。