AVRアセンブリ言語3 レポート

- 実験日：2024年12月05日
- 実験者：2I44 吉髙 僚眞

実験目的

- クロス開発環境を使用できる
- アセンブラ言語（分岐命令）について説明できる

実験内容

- Microchip StudioでATtiny2313のシュミレーション環境を使ってアセンブリの命令を行った際のレジスタの値の変化やプログラムカウンタの動作、ステータスレジスタの動作等を確認する

基本問題

1. ワーク２において、“RCALL KASAN” を“RJMP KASAN” に変更し動作を確認せよ。なぜそのような結果となるかプログラムを示した上で考察せよ。

   動作

   “RCALL KASAN”の場合はRETでLOOP内で呼び出された場所に戻ったが、“RJMP KASAN”の場合は、プログラムの一番初めまで戻った

   考察

   RCALL命令はサブルーチンを呼び出して、RET命令により戻るための命令だが、RJMPはサブルーチンへ飛ぶだけで、戻る呼び出しもとに動作ができない命令だと考えられる。

2. ワーク２において、“KASAN”におけるRET命令を削除し、変更し動作を確認せよ。なぜそのような結果となるかプログラムを示した上で考察せよ。

   プログラム(抜粋)

```
START:
    LDI R16, low(RAMEND) ;サブルーチンを使うための命令
    OUT SPL, R16         ;サブルーチンを使うための命令
    LDI R16, 0x01
    LDI R17, 0x02
LOOP:
    RCALL KASAN
    RCALL KASAN2
    RJMP LOOP
KASAN:
    ADD R16, R17
KASAN2:
    ADD R16, R17
    RET
```

動作

1. 一度目のRCALLでKASANが呼び出されてKASAN2内のRETまでそのままプログラムが実行される。
2. KASAN2内のRETで一度目のRCALL(KASANを呼び出した)位置まで戻る。
3. RCALL KASAN2が呼ばれて、RETまでのプログラムが実行される
4. RETにより、RCALL(KASAN2が呼ばれた)位置まで戻る。

考察
RETではどこのサブルーチンにあるかはかかわらず、RCALLであれば呼ばれた位置まで戻る命令であると考えられる。

発展問題

3. ワーク５において、INC R16をLDI R17,1、ADD R16,R17に変更し、なぜそのような結果となるかプログラムを示した上で考察せよ。

   プログラム(抜粋)

```
START:
    LDI R16, 0xFF
LOOP:
    LDI R17, 1
    ADD R16, R17
    BRBC 1, LOOP
END:
    RJMP END
```

   考察
   ADD R16, R17で0xFF+0x01が行われ、キャリーが発生して値が0になるため、Zフラグが1になる。そのため、BRBCでLOOPに分岐せず、END内のRJMP ENDが次に実行される。

4. ワーク６において、LDI R16,0x01 をLDI R16,0x02に変更し、なぜそのような結果となるかプログラムを示した上で考察せよ。

   プログラム(抜粋)

```
START:
    LDI R16, 0x02
LOOP:
    DEC R16
    BRBS 1, LOOP
END:
    RJMP END
```

   考察
   DEC R16で0x02がデクリメントされるため、Zフラグが0になる。そのため、BRBCでLOOPに分岐せ

ず、END内のRJMP ENDが次に実行される。

AVR入門1

基本問題

1. ウオッチドッグドタイマーについて、調査し報告せよ。 MCUのプログラムが何かの原因で暴走・停止してしまい、電子機器の思わぬ動作をさけるため、MCUが正常に動作しているかを常に監視するのが役割である.

2. 疑似命令について説明せよ。 プログラムをアセンブルする際の指示を定義する命令である。アセンブラの命令とは違いAVRを動作させる命令でない。

感想

一つ一つの命令は簡単だが、組み合わせていくことで、タイマーなどの実装もできると分かった。 PINX, PORTXを間違えてプログラムを書いていたので、そこを間違えないようにしたい。