

割り込み制御プログラミング

要旨

コンピュータにおいて、複数の仕事を並列に処理したり、緊急事態に対処するために用いられている**割り込み制御**を、仮想コンピュータ CASL のシミュレータを用いてその動作を体験し、理解する。

1 概要

通常の**ノイマン型**のコンピュータでは、プログラムを実行する CPU では、1 命令を次々と読み込み処理が加えられる。しかし、複数の処理を切り替えながら並行して実行したり、外部からの要因に対して直ちに別な処理にプログラムを切り替える必要がある場合、**割り込み**という手法が用いられる。

もし割り込みの機能が無いのなら、

- 複数の処理を並行して実行するには、プログラマは小さな処理にプログラムを分割し、それを切り替えるため、複雑になる。
- 複雑な処理中に、外部要因に対して反応するには、**その要因の発生タイミングが不確定**であれば、
 - － 反応までのレスポンス時間が遅れるか、
 - － レスポンスを早くするには、プログラムを細かく分割し、処理が複雑になる。

といった問題が生じる。

そこでノイマン型のコンピュータでは、外部からの処理の切り替え要求がきた時点で、

1. 処理中のプログラムの情報を保存し、
2. 指定された処理に切り替え、
3. 処理の実行後に、本来の処理に復帰する。

といった処理が行われる。

2 割り込み処理

通常の CPU では、複数の**割り込み要因**が存在し、それぞれの要因に応じて、異なった**割り込み処理**が行われる。また、割り込み処理をするにあたって、さらに重要な処理を実行している場合、割り込み処理の**禁止や許可**、さらにその割り込み処理の**優先順位**も重要となる。

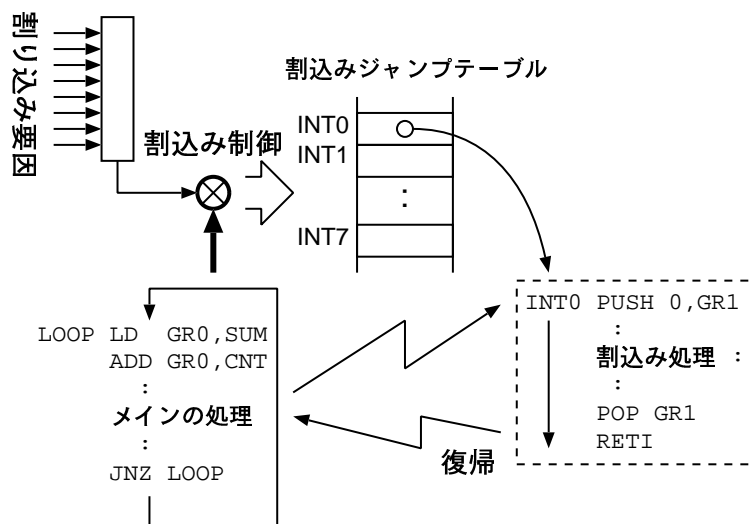


図 1: 割り込みの概念

以下に、今回の実験で使用する CASL シミュレータの割り込み機能の仕様を述べる。

(提出レポートでは、以下の**割り込み仕様**は、記載しなくて良い)

CASL では、以下の割り込みが使用できる。

8つの外部割り込み 外部のハードウェアと接続し、割り込み0～割り込み7までが使用可能。

割り込み0から順に、処理の優先順位が高い。

内部エラー割り込み CPU の命令で内部エラーが発生した時に、発生する割り込み。

ソフトウェア割り込み プログラム中で、SVC 命令によって、呼び出される。

一般的に OS の特殊な処理の呼び出しに用いられることが多い。

以後、実験で使用する CASL シミュレータの割り込み機能について説明する。¹

¹割り込みの詳細な仕様は、通常アーキテクチャ毎に、異なる。

2.1 割り込みジャンプテーブル

割り込み要因に応じた処理のアドレスは、表 1 に示すメモリにあらかじめ登録する必要がある。

割り込みが発生した場合は、CPU はその割り込み要因に応じて、アドレス#0006 ~#000Fの中から 割り込み処理の番地 を読みだし、その先にある一連の処理を実行する。

割り込み要因	メモリアドレス
SVC 命令	#0006
内部エラー	#0007
割り込み 0	#0008
:	
割り込み 7	#000F

```

; 割り込み処理の登録例
;
; 割り込み 0 に INTO を登録
LEA GR0 , INTO
ST GR0 , 8
;
; 割り込み 1 に INT1 を登録
LEA GR0 , INT1
ST GR0 , 9

```

表 1: 割り込みジャンプテーブルと、割り込み処理の設定例

2.2 割り込みマスクと、割り込みの禁止・許可

割り込みにより複数の処理を切り替えることが可能となるが、重要な処理の実行中には、割り込み処理が問題となる場合がある。このため、割り込み処理を実際に呼び出すかどうかは、禁止や許可の設定ができる。

まず、割り込み自体の許可・禁止は、EI,DI 命令で設定され、その状態は、フラグレジスタ (FR: 図 2 参照) に保存される。

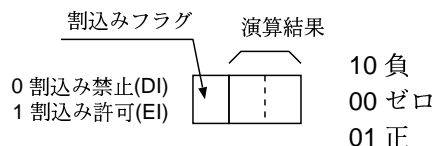


図 2: フラグレジスタ

さらに、複数の割り込みの中から、必要な割り込みを選択できる。マスクレジスタ (MR: 図 3 参照) の対応 bit が 0 の割り込みだけが、割り込み許可される。

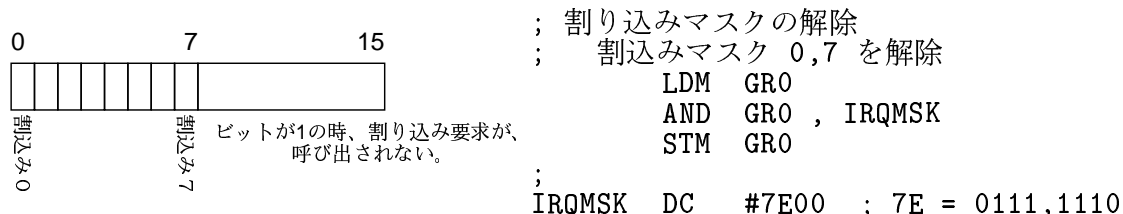


図 3: マスクレジスタと設定プログラム例

2.3 割り込み時の処理と、復帰処理

割り込み要求が発生した場合、次のような一連の処理が CPU 内部で実行命令サイクルに引続き行われる。

1. MR や FR による割り込みの許可 / 禁止の確認
2. FR,GR0 をスタックに保存
3. 割り込みの禁止
4. PC のスタックへの保存
5. 割り込みジャンプテーブルから、割り込み処理アドレスの読みだし
6. 割り込み処理アドレスを PC へ、設定。

また、一連の割り込み処理から、直前に実行していた処理に復帰するには、RETI 命令が利用される。

2.4 割り込みプログラムの例

CASL では、以上のような割り込みの仕様を踏まえ、簡単な割り込みプログラムの例を、表 2 に示す。

```

;          割り込みサンプルプログラム
;          割り込み 0 が発生するまで繰り返す。
;
;          START  PROG
IRQMSK  EQU   #7F00          ; 7F = 0111,1111,0000,0000
PROG    DI
        LEA   GR0 , INTO
        ST    GR0 , 8        ; 割り込み 0 の処理 = INTO
;
        LEA   GR0 , IRQMSK
        STM   GR0          ; MR の割り込み 0 を許可
        EI
;
        LEA   GR0 , 0
        ST    GR0 , FLAG    ; FLAG = 0
;
LOOP    LD    GR1 , FLAG
        LEA   GR1 , 0 , GR1
        JZE  LOOP          ; IF FLAG == 0 THEN GOTO LOOP
HALT    EXIT
;
INTO    LEA   GR0 , 1        ; FLAG = 1
        ST    GR0 , FLAG
        RETI
;
FLAG    DS    1
        END
```

表 2: サンプルプログラム

3 実験内容

1. 付録 A に示すプログラムをいくつか入力し、シミュレータの使用法と、CASL のプログラムの動作を確認せよ。
2. 割込みプログラム例 (表 2) を入力し、動作を確認せよ。
3. 前プログラムを修正し、割込み 0 の発生回数を数えるようにせよ。
(できれば、付録のプログラム PRTHEX を利用し、その回数を表示する)
4. さらに修正を加え、2 つの割込みを用いて、以下の機能を実現せよ。
 - 割込み 1 の発生した回数を数える。
 - 割込み 0 が発生したらプログラムを停止。
5. ソフトウェア割り込みで、付録のプログラム PRTHEX が呼び出されるようにして、課題 3 のプログラムを実現せよ。

4 考察

1. 一般的なプロセッサでは、割り込み要因として「どのようなものが」、「何のために用意されているか」、を文献を調査して答えよ。
2. 一般的に、割り込みが発生した時に CPU のハードウェアにて行われる**一連の処理** (参考 2.3 節) について、以下の点を踏まえて具体的に説明せよ。
 - どういった順序で、どういう処理が行われるか
 - それぞれの処理が行われる理由
 - なぜ、その順序で行われるのか
3. 以下の 2 つの機能を、割り込みを用いて効率良く実現したい。これらに必要な処理について、実現方法を説明せよ。
(選択)
ただし、以下の点について具体的に述べること。
 - どのように割込みを使用するのか?
(割込みの種類は?、割込みの信号をどの様に接続するか?)
 - 割込みの発生するタイミングは
 - 割込みが発生した時に行われる処理は?
 - (a) 一般的に、計算結果をプリンタに出力しながら動く、プログラムがある場合、プリンタの印刷完了を待ちながらでは処理時間が多くかかってしまう。そこで、計算プログラムと印刷処理が、できるだけ同時に並行して行われるにはどのような方式が取られるか、説明せよ。
 - (b) 複数の独立したプログラムに CPU を一定時間ずつ循環配分して、見かけ上プログラムが並行動作する**時分割多重プログラム方式** について、前考察と同じように、具体的に説明せよ。

4. 以下の2つの機能を、仮想コンピュータ CASL を用いて実現したい。具体的なアセンブラによるプログラムリストを交えながら、説明せよ。
(選択)

(a) 以下のような1桁の電子ルーレットを作りたい。

- プログラム開始で0.2秒毎に数値が増える。
- 0.2秒毎にタイマ割り込み(割り込み0)が発生する。
- 停止ボタンを押す(割り込み1)と、ルーレット停止。
- 出力ポート0に出力された値が、表示される。

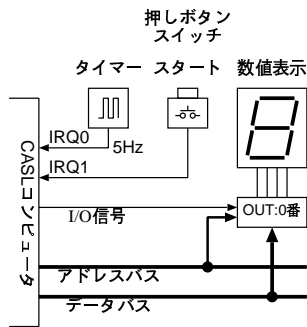


図 4: 1 桁の電子ルーレット

(b) 送信用と受信用の2つのデータ線を持つ通信路で、データ通信を行いたい(全2重通信路)。

- 出力ポート0に出力すると、相手側の入力ポート1にデータが伝わる。相手側にREQ信号が伝わると、割り込み0が発生。
- 相手側からデータが送られてきて、入力ポート1からデータを読み出すと、相手側にACK信号が送られ、割り込み1が発生。

このような割り込みや入出力の状態では、以下のプログラムについて具体的に述べよ。

- アドレス SENDSTR 番地から、SENDLEN 個のデータを送信する。
- さらに送信処理中に、相手側からデータを受信したら、アドレス RECVSTR 番地に記憶し、その受信ワード数を RECVLEN 番地に記憶する。

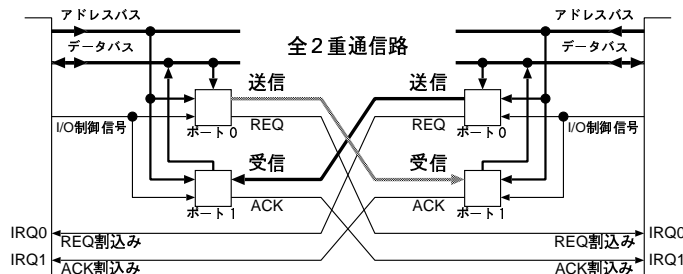


図 5: 全2重通信

繰り返しのあるプログラムの例

```

; START PROG ; 1 ~ N までの加算
;
; PROG LD GRO , ZERO
; ST GRO , SUM ; SUM = 0
; LD GRO , N
; ST GRO , CNT ; CNT = N
;
; LOOP LD GRO , SUM
; ADD GRO , CNT
; ST GRO , SUM ; SUM = SUM + CNT
;
; LD GRO , CNT
; SUB GRO , ONE
; ST GRO , CNT ; CNT = CNT - 1
; JPZ LOOP ; IF CNT >= 0 THEN GOTO LOOP
;
; EXIT ; プログラム終了
;
; ZERO DC 0 ; 定数 0
; ONE DC 1 ; 定数 1
;
; N DC 10 ; 加算する範囲
; CNT DS 1 ; カウンタ変数
; SUM DS 1 ; 加算結果の変数
;
; END

```

関数 (サブルーチン) の呼び出しと、出力命令の例

```

; START PRTHEX ; 16 進の数値表示を呼び出す。
;
; LEA GRO , 100
; CALL PRTHEX ; '64' が表示される
; EXIT
;
; PRTHEX ST GRO , TEMP ; GRO(0 ~ 255) の内容
; を ;
; 関数 ; 16 進数で表示する
;
; LD GR1 , TEMP
; AND GR1 , 15
; LD GRO , TABLE , GR1 ; GRO = TABLE[ N & 15 ]
; ST GRO , STR2
;
; LD GR1 , TEMP
; SRL GR1 , 4
; AND GR1 , 15
; LD GRO , TABLE , GR1 ; GRO = TABLE[ (N/16) & 15 ]
; ST GRO , STR1
;
; OUT STR1 , LEN ; 結果を出力装置に印刷
; RET
;
; TABLE DC '0123456789ABCDEF'
; TEMP DS 1
; STR1 DS 1
; STR2 DS 1
; LEN DC 2
;
; END

```