

17. DSPの操作とデータ転送

DSPに対するCPUの処理は2つである。1つはDSPのスタートおよびストップを行い、もう1つはDSPでの演算に必要なデータの出力および演算結果の入力、すなわちデータ転送である。

[DSPの操作]

CPUがDSPのプログラム領域（共有メモリ）にプログラムやデータを書き込む場合には、DSPをストップし、その後DSPをスタートする必要がある。また、後述するデータ転送を行う時にもDSPのスタート、ストップを行わなければならない。DSPの動作はCRのビットb0とb6を操作して行う。

DSPのストップは次のようにCPUアセンブラプログラムを実行する。

MOV	DX,02D2H	;CRのI/Oアドレス
IN	AL,DX	;CRからビット内容取り込み
OR	AL,40H	;b6=1 (DSPホールド設定)
AND	AL,0FEH	;b0=0 (DSPストップ設定)
OUT	DX,AL	;CRへ出力

DSPのスタートは次のようにCPUアセンブラプログラムを実行する。

MOV	DX,02D2H	;CRのI/Oアドレス
IN	AL,DX	;CRからビット内容取り込み
AND	AL,0BFH	;b6=0 (DSPノーマル設定)
OR	AL,01H	;b0=1 (DSPスタート設定)
OUT	DX,AL	;CRへ出力

[データ転送]

DSPでの演算に必要なデータをCPUから転送する、あるいはDSPで行った演算の結果をCPUへ転送するには次の2つの方法がある。

1. DPRによるデータ転送
2. CPUから共有メモリへの直接アクセス

いずれの方法でも、CPUとDSPは互いに、いつデータが送られたか、あるいは演算が終了したかを調べてデータの書き込みおよび読み出しを行う必要がある。このような場合にはハンドシェイク（Handshake）と呼ばれる通信手段によって、それぞれの動作状況を信号（ビット操作）で知らせるということを行う。具体的には、「データを送った」、「データを受け取った」という情報を「CR」の特定ビットによって表示し、このビットを検出して互いにデータのやり取りを行う。

方法1のDPRによるデータ転送では、DSP側のメモリ構成を意識する必要がないが、複数のデータを転送する場合には1データ毎にハンドシェイクが必要であり、結果として命令のステップ数が増加して実行速度の低下となる。本制御プログラムでは割り込み処理

を行っているため、方法2の共有メモリによるデータ転送を利用しており、DPRはデータ転送の終了合図として使用している。したがって、CPUとDSPはDPRからデータが送信されたかどうかだけを調べればよいので、CPUの「CRのビット4 (b4)」およびDSPの「CRのビット0 (b0)」を使用している。

CPUがDPRへデータを出力する時は、DSPをスタートし、その後

MOV	DX,02D4H	;DPRのI/Oアドレス
MOV	AX,データ	;データをAXレジスタに格納
OUT	DX,AX	;CRへ出力

を実行する。DSPはDPRへデータが送られたかどうかを調べるために

LACK	1	;ACCに1を格納
WAIT:	IN	BUFF,PA0 ;CRの内容を変数BUFFに取り込む
	AND	BUFF ;ACCと変数BUFFとのAND
	BZ	WAIT ;ACCが0ならばWAITへジャンプ
	IN	VAR1,PA1 ;DPRから変数VAR1へデータ取り込み

を実行する(命令は次章を参照)。なお、CPUからDPRへデータを転送しない時にはこれらの処理は必要ない。

一方、DSPがDPRへデータを出力する時は、

OUT	VAR1,PA1	;変数BUFFの内容をDPRへ出力
-----	----------	-------------------

を実行し、CPUはDPRへデータが送られたかどうかを

MOV	DX,02D2H	;CRのI/Oアドレス
DWAIT:	IN	AL,DX ;CRの内容をALレジスタに取り込む
	AND	AL,10H ;ALレジスタと定数10HとのAND
	JZ	DWAIT ;ALレジスタが0ならばDWAITへジャンプ
MOV	DX,02D4H	;DPRのI/Oアドレス
IN	AX,DX	;DPRからAXレジスタへデータ取り込み

を実行して調べ、データを取り込んだ後に、DSPをストップする。

DPRによって複数のデータを転送する場合には、相手がDPRからデータを取り込んだかどうかを調べるために、CRの「BSY」ビットを調べる必要がある。

CPUが共有メモリへのアクセスを行うには、DSPのストップ状態でデータ転送命令を実行する。前述したように、DSPのプログラムメモリ(ウィンドウ0のDSPアドレス0000H~7FFFH)はCPUの物理アドレス「C0000H~CFFFFH」すなわちセグメント「C000H」、オフセット「0000H~FFFFH」上に存在する共有メモリ構成となっている。このDSPプログラムメモリ上にデータ転送のための領域を確保し、その領域を用いてCPUとの間でデータのやり取りを行う方法が「共有メモリによるデータ転送」である。

CPUからDSPのプログラム領域に定義したデータ保存場所にデータを転送するには、まず、DSPのデータアドレスを決定しておく必要がある。通常、DSPのプログラムはDSPアドレスの0番地から記述するが、下位の32ワードはシステムで予約されているため使用できない。したがって、転送用データ領域はプログラムメモリの「50番地」

以降から定義する。DSPアドレス「50番地」はCPUの物理アドレスの「C0064H と C0065H」に対応し、これはセグメント「C000H」オフセット「0064H と 0065H」である。よって、このDSPプログラムメモリにCPUからデータを書き込むには、次のようにデータ転送命令でプログラミングし、その後DSPをスタートさせる。

MOV	AX,0C000H	;共有メモリのセグメント
MOV	ES,AX	;ESセグメント設定
MOV	AX,データ	;AXレジスタにデータ格納
MOV	BX,100	;オフセット (MOV BX,64H)
MOV	ES:[BX],AX	;データ転送

あるいは、ブロック転送命令を使用して、CPUのデータ領域に定義した「SDAT」の内容を転送する場合には

MOV	SI,OFFSET DGROUP:SDAT	;変数 SDAT のオフセット
MOV	AX,0C000H	;共有メモリのセグメント
MOV	ES,AX	;ESセグメント設定
MOV	DI,100	;DIレジスタにDSPデータのオフセット設定
MOVSW		;[DS][SI] [ES][DI]

のように記述する。複数のデータ転送の時は、「MOVSW」をデータ数だけ実行するが、CPUとDSPのデータ定義は連続に対応させておかなければならない。

また、DSPプログラムメモリからデータを読み込むには、DSPをストップし、その後、次のようにデータ転送命令を実行する。

MOV	AX,0C000H	;共有メモリのセグメント
MOV	ES,AX	;ESセグメント設定
MOV	BX,100	;オフセット (MOV BX,64H)
MOV	AX,ES:[BX]	;データ転送

あるいは、ブロック転送命令を使用して、CPUのデータ領域に定義した「RDAT」へ内容を転送する場合には

MOV	AX,DS	
MOV	ES,AX	;ES=DS
MOV	AX,0C000H	;共有メモリのセグメント
MOV	DS,AX	;DSセグメント設定変更
MOV	SI,100	;SIレジスタにDSPデータのオフセット設定
MOV	DI,OFFSET DGROUP:RDAT	;変数 RDAT のオフセット
MOVSW		;[DS][SI] [ES][DI]
MOV	AX,ES	
MOV	DS,AX	;DSレジスタの内容を復帰

のように記述する。複数のデータ転送の時は、「MOVSW」をデータ数だけ実行するが、CPUとDSPのデータ定義は連続に対応させておかなければならない。

この方法では、CPUとDSPプログラムで定義したデータアドレスが完全に一致しておく必要があるため、アドレスには十分注意しなければならない。

なお、DSPにおけるプログラムメモリ（共有メモリ）へのアクセスは次章で説明する。