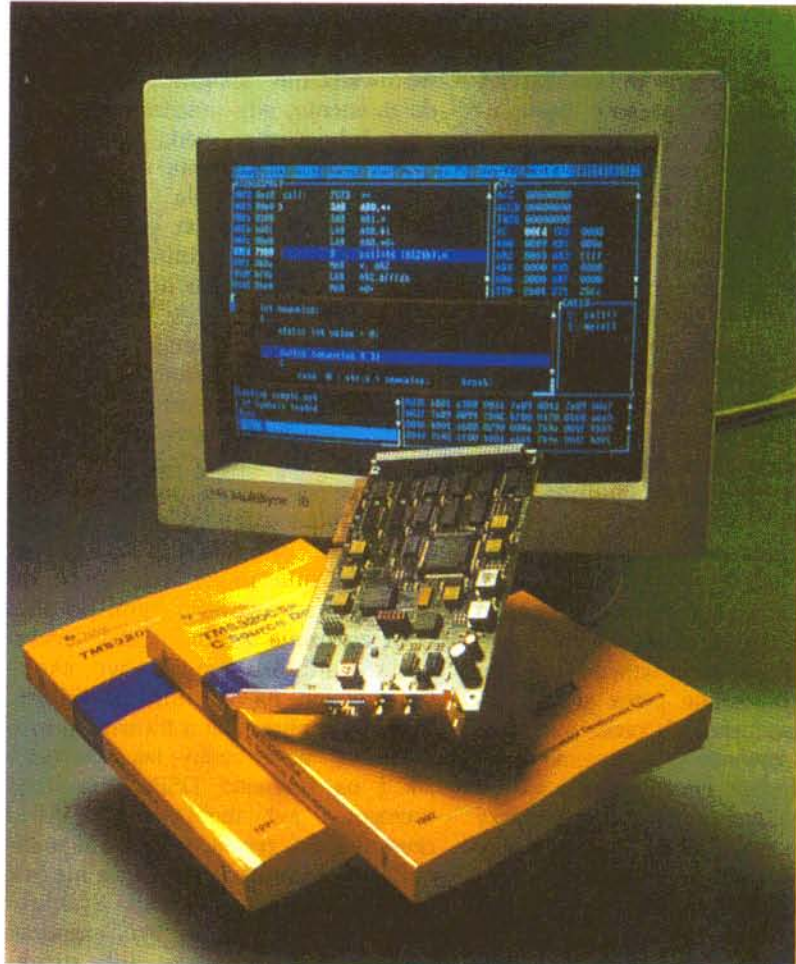


# Procesor sygnałowy TMS 320C25

Po szczegółowym przedstawieniu w poprzednim numerze EP (7/93) architektury układu TMS 320C25: zarządzania pamięcią, jednostki centralnej i schematu funkcjonalnego, w bieżącym numerze kończymy omówienie tego procesora wyszczególnieniem listy instrukcji i przytoczeniem kilku przykładów konfiguracji aplikacyjnych. Dla jasności opisu kontynuujemy numerację rysunków zgodnie z pierwszą częścią.



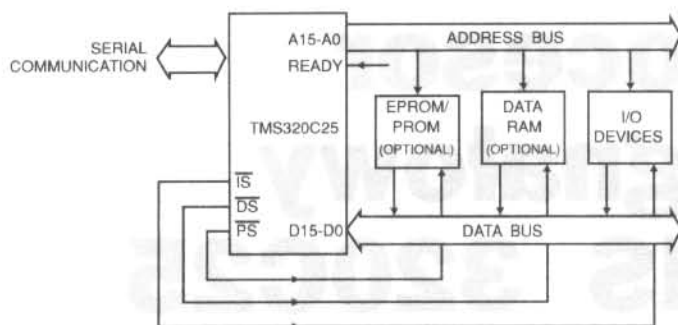
## Sterowanie systemu

Sterowanie pracą procesora TMS 320C25 jest możliwe przy pomocy zegara wewnętrznego, licznika powtórzeń, trzech maskowalnych przerw zewnętrznych użytkownika oraz przerw wewnętrznych, generowanych przez port szeregowy, lub przez czasomierz.

Czasomierz (TIM) w TMS 320C25 odlicza czas w sposób ciągły zgodnie z zegarem CLKOUT1. Przerwanie czasomierza (TINT) powstaje, gdy stan jego osiąga zero. Czasomierz osiąga-

jący zero jest automatycznie ładowany ponownie przez zawartość rejestru „okres” (PRD). W ten sposób może być programowane przerwanie pojawiające się okresowo co  $(PRD \times 1) \times CLKOUT1$ . Funkcja ta przydaje się do kontroli operacji i do synchronizacji próbkowania lub generacji sygnału.

Rozkaz powtarzania operacji umożliwia kolejne wielokrotne (do 256 razy) wykonywanie tej samej instrukcji. Licznik powtórzeń (RPTC) może też zostać załadowany zarówno daną z pamięci (jeśli jest użyty



Rys. 9.

rozkaz RPT), jak i wartością natychmiastową, wziętą z pamięci programu (jeśli jest użyty rozkaz RPTK). Możliwe jest powtarzanie następujących instrukcji: mnożenie z akumulacją, przenoszenie bloków, przenoszenie wejść/wyjść, oraz odczyt i zapis tablic. Te instrukcje, które normalnie są wykonywane w ciągu wielu cykli DSP, są „potokowane” przy powtarzaniu, a czas ich wykonywania zostaje zredukowany do jednego cyklu DSP. Na przykład instrukcja odczytu tablicy TBLR zajmuje zwykle cztery cykle DSP. Jeżeli jest powtarzana, odczyt komórek pamięci odbywa się w rytmie jednego na cykl DSP.

Trzy maskowalne przerwania zewnętrzne (INT0 - INT1 - INT2) skierowują wykonywany proces do innych zadań, według potrzeb użytkownika. Przerwania wewnętrzne mogą być generowane przez port szeregowy, czasomierz, lub przez instrukcje przerwania zawarte w programie. Każdej z nich jest przypisywany odpowiedni poziom priorytetu. Najwyższy priorytet ma RESET, a najniższy port szeregowy. Dwa rejestry stanu ST0 i ST1 wyznaczają tryby i warunki działania DSP. Rejestry te w trakcie wykonywania podprogramu lub przerwania mogą być nadzorowane przez pamięć specyficznymi instrukcjami.

**Interfejsy wejścia/wyjścia**

Procesor TMS 320C25 jest zdolny do sprzęgania z bardzo dużym obszarem adresowym dzięki oddzielnym polom adresowym, tj. polu programowemu, polu danych i polu wejść/wyjść. Wejścia/wyjścia są traktowane w taki sam sposób jak dane. Użycie magistrali danych i adresów zewnętrznych jest zarządzane w DSP identycznie dla danych z pamięci i wejść/wyjść.

Sprzęganie urządzeń wewnętrznych i powolnych pamięci zewnętrznych

jest możliwe dzięki istnieniu wejścia READY, umożliwiającego wprowadzanie stanów oczekiwania do momentu, gdy urządzenie zewnętrzne lub pamięć będą gotowe do zakończenia przesyłania danych.

Ograniczenie pola wejść/wyjść do szesnastu adresów, pozwala wykorzystywać do zarządzania wejściami/wyjściami DSP tylko adresy A0 do A3.

Szeregowy port wejść/wyjść może służyć do łączenia takich urządzeń jak CODEC, przetworniki analogowo-cyfrowe z szeregowym interfejsem czy inne układy szeregowy. Dla realizacji sprzężenia niezbędne jest użycie tylko kilku elementów zewnętrznych. Interfejs ten w żadnym wypadku nie może być użyty do transmisji szeregowych typu RS 232. Dwa związane z nim rejestry (wysyłania i odbioru) mogą być skonfigurowane na 8 lub 16 bitów. Sygnał synchronizacji transmisji może zostać wytworzony tak wewnątrz jak i na zewnątrz DSP. Transmisja szeregowy jest możliwa do 5MHz częstotliwości zegara, przy czym minimalna częstotliwość działania nie jest ograniczona ( $f_{min} = 0Hz$ ). Synchronizacja rastru może być, zgodnie z bitem FSM aktywna lub nieaktywna. Taki sposób działania, z synchronizacją rastru, jest bardzo uży-

teczny w zastosowaniach Telecom typu MIC (G711/712/772 CCIT, lub T1 AT&T).

**Konfiguracja systemu**

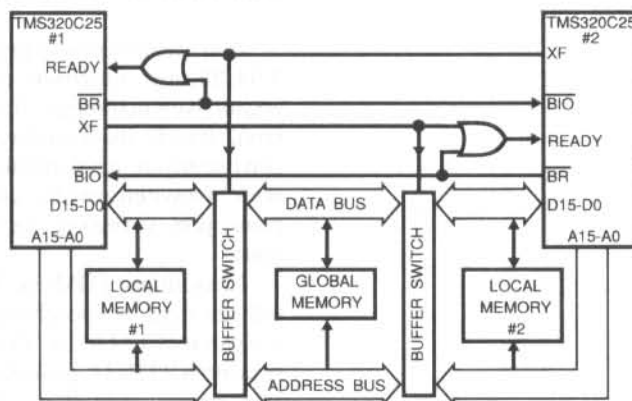
Uniwersalność procesora TMS 320C25 daje możliwość szerokiego zakresu wykorzystania go, przykładowo biorąc . może być stosowany jako:

- monoprocesor (to jest pojedynczy procesor z wewnętrznymi pamięciami 4kB ROM/EPROM i 544/1568B RAM),
- multiprocesor dzielący się pamięcią danych z innymi procesorami,
- koprocesor główny, lub podrzędny, za pośrednictwem przewidzianego do tego celu sprzętowego interfejsu.

Przy zastosowaniu TMS 320C25 jako monoprocesora, sprzęganie z urządzeniami zewnętrznymi odbywa się za pomocą równoległej 16 bitowej magistrali danych, równoległej 16 bitowej magistrali adresowej, trzech sygnałów wyboru używanego pola pamięci (programu, danych, wejść/wyjść) oraz kilku sygnałów kontrolnych.

Na rys. 9 do minimalnego układu monoprocesora dodano zewnętrzne pamięci RAM i ROM albo PROM. Sygnał READY pozwala generować stany oczekiwania potrzebne w przypadku sprzęgania DSP z dowolnymi urządzeniami zewnętrznymi i pamięciami zewnętrznymi bezpośrednio sterowanymi przez TMS 320C25. Mimo to trzeba dodać jeden, lub więcej portów zewnętrznych. Interfejs komunikacji szeregowy pozwala dołączyć CODEC, przetwornik analogowo-cyfrowy lub cyfrowo-analogowy albo też inne urządzenie wchodzące w skład konfiguracji multiprocesora.

W zastosowaniach multiprocesor-



Rys. 10.

rowych procesor TMS 320C25 umożliwia adresowanie pamięci wspólnej dla procesorów i zarządzanie w niej swoim udziałem za pomocą sygnałów BR (Bus Request) i Ready.

8 bitowy wewnętrzny rejestr adresowania wspólnej pamięci (GREG) pozwala sięgać w pamięci danych TMS 320C25, w obszar do 32kB, we wspólnej pamięci. Zawartość tego rejestru wyznacza wielkość wspólnej pamięci. Jeśli wykonywana instrukcja ma zaadresować argument operacji we wspólnej pamięci, automatycznie jest generowane żądanie dostępu do magistrali (BR). Sygnał READY przybiera wtedy wartość 1 i generuje stany oczekiwania przedłużając bieżący cykl, umożliwiając DSP dostęp do wspólnej pamięci.

Przy multiprocesorowej konfiguracji systemu, pola adresowe każdego z procesorów są podzielone na dwie części, pamięć lokalną - dostępną jedynie dla własnego procesora i pamięć wspólną - dostępną dla wszystkich procesorów. Ta ostatnia pozwala na obróbkę wspólnych dla danego zastosowania danych.

**Rys. 10** pokazuje równoległą konfigurację systemu, oddającą część pamięci do wspólnego użytku. Oba układy DSP dzielą wspólną pamięć, ale wykonują własne programy. O dostępie do wspólnej pamięci rozstrzyga się sprzętowo, za pomocą sygnałów XF, BR, BIO i READY. Gdy przez żądanie dostępu do pamięci globalnej zostaje uaktywniony sygnał BR w jednym DSP, sygnał BIO wywołuje sygnał skoku BIOZ w drugim DSP, który musi zostać wykonany i zwolnić magistralę, ustawiając odpowiedni wskaźnik (flag) na wyjściu XF. Sygnał READY w DSP żądającym dostępu do pamięci przybiera wtedy wartość 1, i odczyt (lub zapis) w pamięci może nastąpić bez konfliktu.

Multiprocesorowe działanie TMS 320C25 może zostać również implementowane przez użycie sygnałów HOLD i HOLDA, którym towarzyszą przerwania. Sygnały HOLD i HOLDA (żądanie zawieszenia działania DSP i jego wznowienia) pozwalają innemu procesorowi odczytywać lub zapisywać w pamięci zewnętrznej TMS 320C25. Adresowanie pamięci danych i programu jest więc możliwe, ponieważ TMS 320C25 po-

ACCUMULATOR MEMORY REFERENCE INSTRUCTIONS						
Mnemonic and Description		Words	16-Bit Opcode			
			MSB		LSB	
ABS	Absolute value of accumulator	1	1100	1110	0001	1011
ADD	Add to accumulator with shift	1	0000	SSSS	IDDD	DDDD
ADDCC*	Add to accumulator with carry	1	0100	0011	IDDD	DDDD
ADDH	Add to high accumulator	1	0100	1000	IDDD	DDDD
ADDK**	Add to accumulator short immediate	1	1100	1100	KKKK	KKKK
ADDS	Add to low accumulator with sign-extension suppressed	1	0100	1001	IDDD	DDDD
ADDT*	Add to accumulator with shift specified by T register	1	0100	1010	IDDD	DDDD
ADLK	Add to accumulator long immediate with shift	2	1101	SSSS	0000	0010
AND	AND with accumulator	1	0100	1110	IDDD	DDDD
ANDK*	AND immediate with accumulator with shift	2	1101	SSSS	0000	0100
CMPL*	Complement accumulator	1	1100	1110	0010	0111
LAC	Load accumulator with shift	1	0010	SSSS	IDDD	DDDD
LACK	Load accumulator short immediate	1	1100	1010	KKKK	KKKK
LACT*	Load accumulator with shift specified by T register	1	0100	0010	IDDD	DDDD
LALK*	Load accumulator long immediate with shift	2	1101	SSSS	0000	0001
NEG*	Negate accumulator	1	1100	1110	0010	0011
NORM*	Normalize contents of accumulator	1	1100	1110	1010	0010
OR	OR with accumulator	1	0100	1101	IDDD	DDDD
ORK*	OR immediate with accumulator with shift	2	1101	SSSS	0000	0101
ROL**	Rotate accumulator left	1	1100	1110	0011	0100
ROR**	Rotate accumulator right	1	1100	1110	0011	0101
SAC*	Store high accumulator with shift	1	0110	1XXX	IDDD	DDDD
SACL	Store low accumulator with shift	1	0110	0XXX	IDDD	DDDD
SBLK*	Subtract from accumulator long immediate with shift	2	1101	SSSS	0000	0011
SFL*	Shift accumulator left	1	1100	1110	0001	1000
SFR*	Shift accumulator right	1	1100	1110	0001	1001
SUB	Subtract from accumulator with shift	1	0001	SSSS	IDDD	DDDD
SUBB**	Subtract from accumulator with borrow	1	0100	1111	IDDD	DDDD
SUBC	Conditional subtract	1	0100	0111	IDDD	DDDD
SUBH	Subtract from high accumulator	1	0100	0100	IDDD	DDDD
SUBK**	Subtract from accumulator short immediate	1	1100	1101	KKKK	KKKK
SUBS	Subtract from low accumulator with sign-extension suppressed	1	0100	0101	IDDD	DDDD
SUBT*	Subtract from accumulator with shift specified by T register	1	0100	0110	IDDD	DDDD
XOR	Exclusive-OR with accumulator	1	0100	1100	IDDD	DDDD
XORK*	Exclusive-OR immediate with accumulator with shift	2	1101	SSSS	0000	0110
ZAC	Zero accumulator	1	1100	1010	0000	0000
ZALH	Zero low accumulator and load high accumulator	1	0100	0000	IDDD	DDDD
ZALR**	Zero low accumulator and load high accumulator with rounding	1	0111	1011	IDDD	DDDD
ZALS	Zero accumulator and load low accumulator with sign-extension suppressed	1	0100	0001	IDDD	DDDD

\*) This instruction is specific to the TMS320C2x instruction set.

\*\*\*) This instruction is specific to the TMS320C25/E25 instruction set.

CONTROL INSTRUCTIONS						
Mnemonic and Description		Words	16-Bit Opcode			
			MSB		LSB	
BIT*	Test bit	1	1001	BBBB	IDDD	DDDD
BITT*	Test bit specified by T register	1	0101	0111	IDDD	DDDD
CNFD***	Configure block as data memory	1	1100	1110	0000	0100
CNFP***	Configure block as program memory	1	1100	1110	0000	0101
CONF***	Configure block as data/program memory	1	1100	1110	0011	11KK
DINT	Disable interrupt	1	1100	1110	0000	0001
EINT	Enable interrupt	1	1100	1110	0000	0000
IDLE*	Idle until interrupt	1	1100	1110	0001	1111
LST	Load status register ST0	1	0101	0000	IDDD	DDDD
LST1*	Load status register ST1	1	0101	0001	IDDD	DDDD
NOP	No operation	1	0101	0101	0000	0000
POP	Pop top of stack to low accumulator	1	1100	1110	0001	1101
POPD*	Pop top of stack to data memory	1	0111	1010	IDDD	DDDD
PSHD*	Push data memory value onto stack	1	0101	0100	IDDD	DDDD
PUSH	Push low accumulator onto stack	1	1100	1110	0001	1100
RC**	Reset carry bit	1	1100	1110	0011	0000
RHM**	Reset hold mode	1	1100	1110	0011	1000
ROVM	Reset overflow mode	1	1100	1110	0000	0010
RPT*	Repeat instruction as specified by data memory value	1	0100	1011	IDDD	DDDD
RPTK*	Repeat instruction as specified by immediate value	1	1100	1011	KKKK	KKKK
RSXM*	Reset sign-extension mode	1	1100	1110	0000	0110
RTC**	Reset test/control flag	1	1100	1110	0011	0010
SC**	Set carry bit	1	1100	1110	0011	0001
SHM**	Set hold mode	1	1100	1110	0011	1001
SOVM	Set overflow mode	1	1100	1110	0000	0011
SST	Store status register ST0	1	0111	1000	IDDD	DDDD
SST1*	Store status register ST1	1	0111	1001	IDDD	DDDD
SSXM*	Set sign-extension mode	1	1100	1110	0000	0111
STC**	Set test/control flag	1	1100	1110	0011	0011

\*) This instruction is specific to the TMS320C2x instruction set.

\*\*\*) This instruction is specific to the TMS320C25/E25 instruction set.

\*\*\*\*) The CONF instruction is specific to the TMS320C26 instruction set; the instructions CNFD and CNFP are undefined.

AUXILIARY REGISTERS AND DATA PAGE POINTER INSTRUCTIONS					
Mnemonic and Description		Words	16-Bit Opcode		
			MSB		LSB
ADRK**	Add to auxiliary register short immediate	1	0111	1110	KKKK KKKK
CMPR*	Compare auxiliary register with auxiliary register ARO	1	1100	1110	0101 00KK
LAR	Load auxiliary register	1	0011	0RRR	IDDD DDDD
LARK	Load auxiliary register short immediate	1	1100	0RRR	KKKK KKKK
LARP	Load auxiliary register pointer	1	0101	0101	1000 1RRR
LDP	Load data memory page pointer	1	0101	0010	IDDD DDDD
LDPK	Load data memory page pointer immediate	1	1100	100K	KKKK KKKK
LRLK*	Load auxiliary register long immediate	2	1101	0RRR	0000 0000
MAR	Modify auxiliary register	1	0101	0101	IDDD DDDD
SAR	Store auxiliary register	1	0111	0RRR	IDDD DDDD
SBRK**	Subtract from auxiliary register short immediate	1	0111	1111	KKKK KKKK

T REGISTER, P REGISTER, AND MULTIPLY INSTRUCTIONS					
Mnemonic and Description		Words	16-Bit Opcode		
			MSB		LSB
APAC	Add P register to accumulator	1	1100	1110	0001 0101
LPH*	Load high P register	1	0101	0011	IDDD DDDD
LT	Load T register	1	0011	1100	IDDD DDDD
LTA	Load T register and accumulate previous product	1	0011	1101	IDDD DDDD
LTD	Load T register, and accumulate previous product and move data	1	0011	1111	IDDD DDDD
LTP*	Load T register and store P register in accumulator	1	0011	1110	IDDD DDDD
LTS*	Load T register and subtract previous product	1	0101	1011	IDDD DDDD
MAC*	Multiply and accumulate	2	0101	1101	IDDD DDDD
MACD*	Multiply and accumulate with data move	2	0101	1100	IDDD DDDD
MPY	Multiply (with T register, store product in P register)	1	0011	1000	IDDD DDDD
MPYA**	Multiply and accumulate previous product	1	0011	1010	IDDD DDDD
MPYK	Multiply immediate	1	101K	KKKK	KKKK KKKK
MPYS**	Multiply and subtract previous product	1	1100	1111	IDDD DDDD
MPYU**	Multiply unsigned	1	0011	1011	IDDD DDDD
PAC	Load accumulator with P register	1	1100	1110	0001 0100
SPAC	Subtract P register from accumulator	1	1100	1110	0001 0110
SPH**	Store high P register	1	0111	1101	IDDD DDDD
SPL**	Store low P register	1	0111	1100	IDDD DDDD
SPM*	Set P register output shift mode	1	1100	1110	0000 10KK
SQRA*	Square and accumulate	1	0011	1001	IDDD DDDD
SQRS*	Square and subtract previous product	1	0101	1010	IDDD DDDD

\*) This instruction is specific to the TMS320C2x instruction set.  
 \*\*) This instruction is specific to the TMS320C25/E25 instruction set.

BRANCH/CALL INSTRUCTIONS					
Mnemonic and Description		Words	16-Bit Opcode		
			MSB		LSB
B	Branch unconditionally	2	1111	1111	IDDD DDDD
BACC*	Branch to address specified by accumulator	1	1100	1110	0010 0101
BANZ	Branch on auxiliary register not zero	2	1111	1011	IDDD DDDD
BBNZ*	Branch if TC bit <= 0	2	1111	1001	IDDD DDDD
BBZ*	Branch if TC bit = 0	2	1111	1000	IDDD DDDD
BC**	Branch on carry	2	0101	1110	IDDD DDDD
BGEZ	Branch if accumulator >= 0	2	1111	0100	IDDD DDDD
BGZ	Branch if accumulator > 0	2	1111	0001	IDDD DDDD
BIOZ	Branch on I/O status = 0	2	1111	1010	IDDD DDDD
BLEZ	Branch if accumulator <= 0	2	1111	0010	IDDD DDDD
BLZ	Branch if accumulator < 0	2	1111	0011	IDDD DDDD
BNC**	Branch on no carry	2	0101	1111	IDDD DDDD
BNV*	Branch if no overflow	2	1111	0111	IDDD DDDD
BNZ	Branch if accumulator <> 0	2	1111	0101	IDDD DDDD
BV	Branch on overflow	2	1111	0000	IDDD DDDD
BZ	Branch if accumulator <> 0	2	1111	0110	IDDD DDDD
CALA	Call subroutine indirect	1	1100	1110	0010 0100
CALL	Call subroutine	2	1111	1110	IDDD DDDD
RET	Return from subroutine	1	1100	1110	0010 0110
TRAP*	Software interrupt	1	1100	1110	0001 1110

I/O AND DATA MEMORY OPERATIONS					
Mnemonic and Description		Words	16-Bit Opcode		
			MSB		LSB
BLKD*	Block move from data memory to data memory	2	1111	1101	IDDD DDDD
BLKP*	Block move from program memory to data memory	2	1111	1100	IDDD DDDD
DMOV	Data move in data memory	1	0101	0110	IDDD DDDD
FORT*	Format serial port registers	1	1100	1110	0000 111K
IN	Input data from port	1	1000	AAAA	IDDD DDDD
OUT	Output data to port	1	1110	AAAA	IDDD DDDD
RFSM**	Reset serial port frame synchronization mode	1	1100	1110	0011 0110
RTXM*	Reset serial port transmit mode	1	1100	1110	0010 0000
RXF*	Reset external flag	1	1100	1110	0000 1100
SFSM**	Set serial port frame synchronization mode	1	1100	1110	0011 0111
STXM*	Set serial port transmit mode	1	1100	1110	0010 0001
SXF*	Set external flag	1	1100	1110	0000 1101
TBLR	Table read	1	0100	1000	1DDD DDDD
TBLW	Table write	1	0101	1001	1DDD DDDD

\*) This instruction is specific to the TMS320C2x instruction set.  
 \*\*) This instruction is specific to the TMS320C25/E25 instruction set.

zostaje w trybie HALT, zawieszającym działanie procesora i zwalniającym magistralę. Ten rodzaj działania jest bardzo przydatny do rozpoczęcia i zmieniania programu wykonywanego przez TMS 320C25.

Gdy programy mają być wykonywane w czasie rzeczywistym, i nie mogą być w tym czasie przerywane, wszystkie żądania dostępu do pamięci zostają wstrzymane przez maskowanie przerwań. Jest to konfiguracja typowa dla koprocesorowych zastosowań DSP.

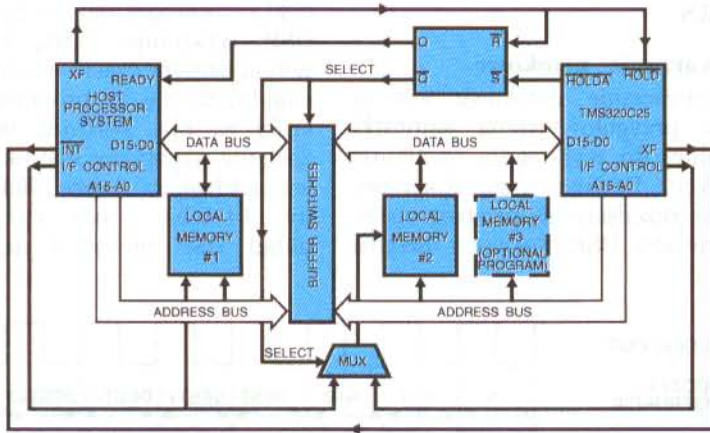
Układ TMS 320C25 został na tyle ulepszony, aby umożliwić działanie w nowym trybie zawieszania wykonywania programu, pozwalającym na wymianę DMA w trakcie wykonywania programu wewnątrz. Wymaga to, aby wykonywany program jak i dane znajdowały się w wewnętrznej pamięci. Program wykonywany jest aż do napotkania instrukcji wymagającej dostępu na zewnątrz, i dopiero wtedy zostaje on zawieszony do chwili udostępnienia magistrali zewnętrznych. Jest to tryb działania bardzo przydatny do zastosowań multiprocesorowych.

Sprzężenie mikroprocesora z zewnętrznymi urządzeniami do obróbki sygnałów w standardzie 16 lub 32 bitowym jest stosowane bardzo często. Takie konfiguracje są użyteczne zwłaszcza w systemach o szerokim zakresie zastosowań. **Rys. 11** pokazuje architekturę systemu z interfejsem sterowanym sygnałami HOLD i HOLDA.

Jedną z zalet układu TMS 320C25 w zastosowaniach multiprocesorowych jest zdolność synchronizacji z zewnętrznymi przebiegami za pośrednictwem sygnału SYNC. Można więc zsynchronizować zegary większej liczby procesorów w TMS 320C25. Procedura ta umożliwi też synchronizację sygnałów zewnętrznych wychodzących z DSP.

### Tryby adresowania i instrukcje TMS 320C25

Lista rozkazów procesora TMS 320C25 zawiera skondensowane operacje obróbki sygnału i takie operacje ogólne, jak sterowanie multiprocesorowe i sterowanie standardowe. Dla uzyskania maksymalnej szybkości działania, bieżąca instrukcja jest wykonywana równocześnie z dekodowaniem następnej i odbiorem kolejnej. Ponieważ ta sama zewnętrzna magistrala danych jest używana do pamięci programu,



Rys. 11.

pamięci danych i wejść/wyjść, to liczba cykli potrzebna do wykonania instrukcji jest zmienna i zależy od następnych i poprzednich instrukcji. Wszystko zależy zatem od sposobu użytkowania magistrali zewnętrznej. Trzeba pamiętać, że DSP osiąga maksymalną sprawność, gdy program i dane są zawarte wewnątrz DSP.

W TMS 320C25 są zaimplementowane trzy tryby adresowania: bezpośredni, pośredni i natychmiastowy. Tryb bezpośredni i tryb pośredni nadają się do adresowania pamięci danych. Przy stosowaniu adresowania bezpośredniego, adres 16 bitowy jest dzielony na dwie części, tak aby pozostawić pole 9 bitów dla wskaźnika strony pamięci danych (DP). Pozostałe 7 bitów pozwala adresować 128B stanowiących stronę wskazywaną przez DP. DP pozwala adresować 512 stron po 128B i umożliwia DSP bezpośrednie zarządzanie 64kB pamięci danych. 7 bitów adresu zawartych w instrukcji adresuje wybraną przez DSP jedną komórkę pamięci strony o 128B. Adresowanie bezpośrednie może być stosowane do wszystkich instrukcji, z wyjątkiem instrukcji CALL, skoku i oczywiście instrukcji stosujących adresowanie natychmiastowe, lub instrukcji bez argumentu.

Tryb adresowania pośredniego jest bardzo skuteczny i elastyczny w stosowaniu, ze względu na przewidziane do tego celu 8 rejestrów pomocniczych (A0 do A7). Adres danej do odczytania (lub zapisania) jest umieszczany w jednym z ośmiu rejestrów pomocniczych, a instrukcja wykonawcza, aby dotrzeć do danej, wprowadza go do magistrali adresów.

Do wybrania jednego z ośmiu rejestrów wystarczy uruchomić wskaź-

nik rejestru pomocniczego ARP wybierając jedną z liczb od 0 do 7. Jednostka arytmetyczna związana z rejestrami pomocniczymi (ARAU) pozwala na wykonywanie operacji arytmetycznych bez znaku na zawartości rejestrów pomocniczych, i to w czasie bieżącego cyklu, z możliwością zmiany wartości wskaźnika (ARP).

Tryb adresowania pośredniego z odwróceniem bitu reorganizuje dane pamięci do bezpośredniej implementacji FFT. Przy zastosowaniu tego trybu kierunek przesuwu sygnału przenoszenia w ARAU zostaje odwrócony i stan rejestru pomocniczego ARO jest dodawany do (albo odejmowany od) bieżącego stanu rejestru pomocniczego. Zastosowanie tego trybu adresowania wymaga jednak inicjalizacji ARO na poziomie  $N/2$  (gdzie N jest liczbą punktów FFT). AR i ARP winny zawierać adres pierwszego punktu, dla którego wykonuje się FFT.

Przy adresowaniu natychmiastowym sama instrukcja zawiera wartość argumentu. Procesor TMS 320C25 zawiera instrukcje natych-

miastowe krótkie (ze stałymi 8 i 13 bitowymi) składające się z jednego słowa i instrukcje natychmiastowe długie (do stałych 16 bitowych) składające się z dwóch słów. W tym ostatnim przypadku, pierwszy bit jest kodem operacyjnym, a drugie słowo stałą (dana natychmiastowa). Zestaw instrukcji TMS 320C25 zawiera 17 instrukcji adresowania natychmiastowego.

W porównaniu z TMS 320C20, do zestawu instrukcji procesora TMS320C25 dodano 24 nowe instrukcje, aby zwiększyć szybkość działania i uprościć jego użycie. Te nowe instrukcje można podzielić na cztery kategorie:

- 1) operacje arytmetyczne o zwiększonej dokładności (ADDC, SUBB, MPYU, BC, BNC, SC, RC),
- 2) operacje filtrowania przystosowanego (MPYA, MPYS, ZALR),
- 3) operacje wejść/wyjść i sterowania (SHM, RHM, STC, RTC, SFSM RFSM),
- 4) instrukcje działań na akumulatorze i rejestrach (SPH, SPL, ADDK, SUBK, ADRK, SBRK, ROL, ROR).

Występujący w tablicach symbol (\*) oznacza instrukcje nie wchodzące w skład instrukcji TMS 320 10, a symbol (\*\*) oznacza instrukcje nie wchodzące w skład instrukcji TMS 320 20.

Dla lepszego zrozumienia obok listy rozkazów w postaci skrótów nazw, podano także ich pełne znaczenie. Na zakończenie warto zasygnalizować, że dla rodziny procesorów TMS 320CXX istnieją programy narzędziowe do obliczeń i opracowywania zastosowań, niektóre z nich zostaną omówione dalej.

Tryb adresowania	Wykonana operacja
OP A	adresowanie bezpośrednie
OP*(,NARP)	pośrednie; AR jest niezmienione
OP*+(,NARP)	pośrednie; bieżący rejestr pomocniczy AR jest zwiększony
OP*-(,NARP)	pośrednie; bieżący rejestr pomocniczy AR jest zmniejszony
OP*O+(,NARP)	pośrednie; ARO jest dodane do bieżącego rej. AR
OP*O-(,NARP)	pośrednie; ARO jest odjęte od bieżącego rej. AR
OP*BRO+(,NARP)	pośrednie; ARO jest dodane do bieżącego rej. AR (z odwróconym przesuwem przenoszenia) (zatrzymanie w pamięci)
OP*BRO-(,NARP)	pośrednie; ARO jest odjęte od bieżącego rej. AR (z odwróconym przesuwem przenoszenia)
NARP = nowy ARP - to pole jest opcjonalne i pozwala załadować nową wartość do wskaźnika rejestru pomocniczego.	

**Moduł obliczania TMS 320C25**

Dla każdego ze swoich procesorów, firma Texas Instruments opracowała zestaw bardzo skutecznych narzędzi do obliczeń i opracowywania algorytmów obróbki sygnałów.

Kit 320C25 zawiera:

- skróconą kartę do PC/AT z procesorem 320C26/40MHz, 64kB szybką pamięć SRAM DATA (danych) i PROGRAMME (programu), 14 bitowy interfejs audio zawierający przetwornik analogowo-cyfrowy i przetwornik cyfrowo-analogowy, połączone wzmacniacze operacyjne z gniazdkami RCA, układ wspomagania emulacji czasu rzeczywistego, złącze emulacji zewnętrznych wejść/wyjść, gniazdo portu szeregowego.

- assembler linker coff obsługujący wszystkie 16 bitowe DSP Texas Instruments.

- źródłowy debugger symboliczny assemblera i C. Tradycyjne funkcje uruchamiania są dostępne poprzez rozwijane menu.

- szczegółowe podręczniki z kompletnymi schematami.

- programy sterujące własnego interfejsu.

Liczne notki aplikacyjne zgromadzono w trzech podręcznikach:

DIGITAL SIGNAL PROCESSING, APPLICATIONS TOME I, II, III.

Kity istniejące to:

320C10, C15, C16:TMDS3260016 (7900 FFR TTC)

320C25, C26:TMDS3260026 (8900 FFR TTC)

320C50, C51, C52, C53:TMDS3260050 (8900 FFR TTC)

320C30, C31:TMDS3260030 (8900 FFR TTC)

Szkoły i uniwersytety mogą uzyskać specjalne zniżki przy zakupie.

Kity są rozprowadzane przez przedstawiciela Texas Instruments:

TEKELEC AIRTRONIC

BP 2

92315 SEVRES CEDEX

Tel. (1) 46.23.24.25

Delanco Spry proponuje karty rozprowadzane przez EXCEM służące do zbierania i obróbki sygnałów, oparte na procesorze 320C25, wyposażone w oprogramowanie narzędziowe potrzebne do opracowania zastosowań, w tym assembler A320, debugger i linker. Programy aplikacyjne są również dostępne w:

EXCEM

29 av. Mary

92500 RUEIL/MALMAISON

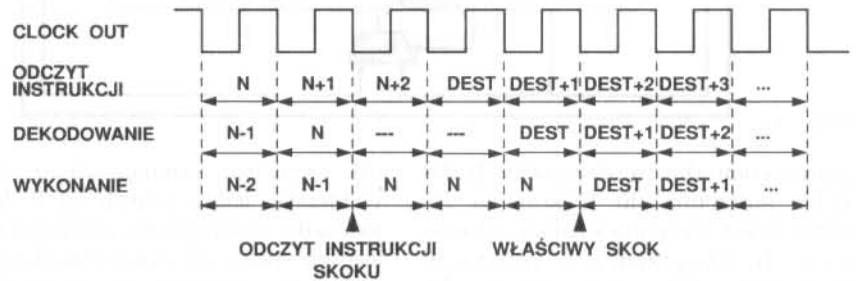
Tel. (1) 47.52.13.44

**ANEKS**

**Przetwarzanie potokowe**

Przetwarzanie potokowe jest to sposób przygotowywania jednostki centralnej do wykonania następnej instrukcji. Podane niżej diagramy czasowe przedstawiają przebieg operacji wewnątrz DSP. Pokazany przypa-

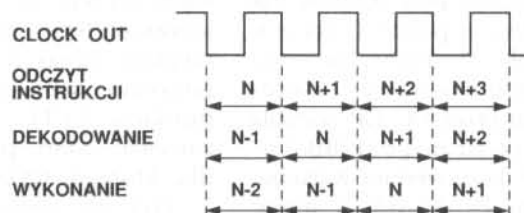
cyklu. Konkretnie, w czasie gdy DSP wykonuje jedną instrukcję, w tym samym czasie odczytuje inną, znajdującą się w programie o dwa kroki za nią. Wskutek tego, jeżeli zostanie napotkana instrukcja skoku, ten mechanizm zostaje silnie zaburzony, i trzeba wielu cykli, aby układ DSP powrócił do swojego



dek dotyczy procesora TMS 320. W trakcie opadającego zbocza sygnału zegara CLKOUT licznik programu (PC) jest ładowany przy adresie wskazującym na instrukcję czytania  $N + 2$  (ładowanie PC), podczas gdy instrukcja  $N + 1$  jest dekodowana, a wykonywana jest instrukcja bieżąca  $N$ . Jest to potok o trzech poziomach. W niektórych przypadkach, jak wykonywanie instrukcji

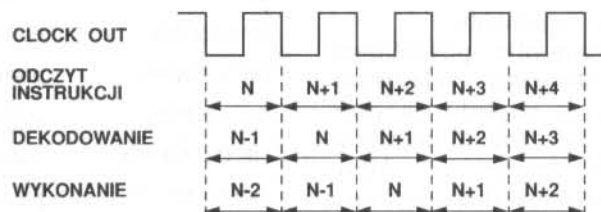
rytmu działania. Uwaga ta jest bardzo istotna w przypadku pisania programu, który ma być wykonywany w czasie rzeczywistym.

Lepiej więc nie używać instrukcji skoku, (a w każdym razie jak najrzadziej), bo można stracić wiele cennych cykli DSP. W pewnych przypadkach jest korzystniejszą częścią programu, która ma być wykonana trzykrotnie (lub więcej razy) nie



na danych pamięci wewnętrznej RAM w DSP, odczyt i dekodowanie może odbywać się w czasie jednego cyklu. Wówczas potok zos-

realizować jako podprogram, ale skopiować właściwą liczbę razy w programie głównym. Wielkość programu nie zawsze bowiem jest



taje ograniczony do dwóch poziomów. Nie zwiększa to w tym stopniu szybkości działania, zegar bowiem nie przyspiesza, gdy jedna instrukcja jest wykonywana w czasie jednego

najważniejsza, a automatyzm rozwiązań należy stosować ostrożnie.

**ERP**