

Co w LPT-cie piszczy

część 3

W ostatniej części krótkiego cyklu artykułów przedstawiamy najbardziej rozwinięty tryb pracy interfejsu Centronics - ECP (Extended Capabilities Port). W stosunku do dwóch wcześniej przedstawionych, ECP jest prawdziwym „kombajnem” możliwości i szybkości, co niestety nieco komplikuje jego obsługę.

Tryby SPP/ECP/EPP od strony praktycznej

Interfejs ECP opracowano w celu zwiększenia jego elastyczności, zapewnienia możliwości dynamicznego modyfikowania kierunku przepływu danych, a także jak najlepszego dostosowania do interfejsu ISA, co wpływa na jego sprzętowe implementacje. W odróżnieniu od standardów SPP i EPP, w ECP użytkownik ma stosunkowo niewielki (bezpośredni) wpływ na stany logiczne na wyprowadzeniach portu - są one obsługiwane sprzętowo przez specjalizowane zaimplementowane w interfejs.

ECP - skomplikowana potęga

Szybkość transmisji danych przesyłanych przez port LPT w trybie ECP jest do 50 razy większa

niż w przypadku SPP, co osiągnięto dzięki zastosowaniu następujących mechanizmów:

- buforów FIFO dla kolejek wejściowych (dane odczytywane przez komputer) i wyjściowych (dane przesyłane z komputera na zewnątrz),
- transferu danych do/z portu za pomocą kanału DMA (*Direct Memory Access*),
- zastosowaniu kompresji przesyłanych danych, która działa on-line.

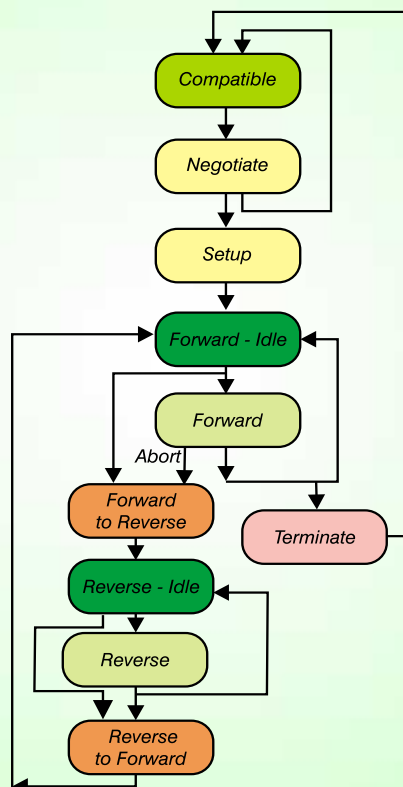
Zastosowany w ECP algorytm kompresji (RLE - *Run Length Encoding*) nie jest zbyt wydajny, ale wymaga niewielkich mocy obliczeniowych i doskonale nadaje się do kompresowania bloków danych składających się z ciągów podobnych znaków, czyli danych typowych dla drukowania w trybie graficznym lub przesyłania obrazów ze skanera lub faksu do PC (i z PC do faksu). Maksymalny stopień kompresji danych wynosi 64:1, w praktyce nie przekracza wartości 58:1.

Twórcy standardu ECP (Hewlett Packard i Microsoft) przewidzieli także możliwość adresowania urządzeń dołączanych do portu, dzięki czemu port równoległy można wykorzystać jako dwukierunkowy uniwersalny interfejs, umożliwiający

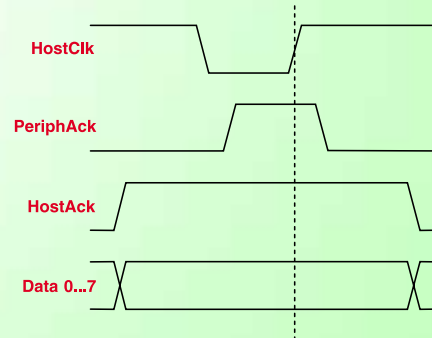
dołączenie do PC wielu urządzeń bez konieczności stosowania dodatkowych kart zwiększających liczbę portów równoległych.

Z założenia port Centronics z trybem ECP musi być kompatybilny z wcześniejszymi postaciami z trybem SPP i EPP. W związku z tym, pomimo znacznie bardziej złożonego protokołu, wykorzystywane są takie same (jak w SPP i EPP) złącza, istnieje także zazwyczaj możliwość przełączenia portu w jeden z tych trybów pracy. W **tab. 5** znajduje się zestawienie sygnałów występujących w ECP wraz z ich przypisaniem do wyprowadzeń złącza. Funkcje sygnałów są następujące:

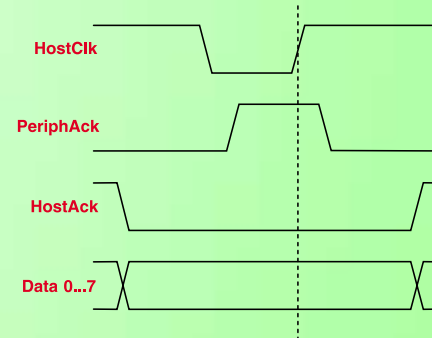
- *HostCLK* - sygnał informujący urządzenie zewnętrzne niskim poziomem, że komputer wystawił dane na *Data 0..7*. Narastające zbocze tego sygnału powinno zatrząskiwac dane w odbiorniku.
- *Data 0..7* - dwukierunkowe linie danych.
- *PeriphCLK* - niski poziom tego sygnału informuje komputer, że na *Data 0..7* są dane do odebrania. Narastające zbocze tego sygnału powinno zatrząskiwac dane w buforze wejściowym interfejsu komputera.
- *PeriphAck* - spełnia podwójną rolę:



Rys. 10.



Rys. 11.



Rys. 12.

Tab. 5. Nazwy wyprowadzeń interfejsu ECP.

Numer styku w DB25	Numer styku w 36-stykowym złączu Centronics	Nazwa sygnału ECP	Kierunek	Dostęp poprzez rejestr
1	1	HostCLK	wy	-
2	2	Data 0	we/wy	Data EPP
3	3	Data 1	we/wy	Data EPP
4	4	Data 2	we/wy	Data EPP
5	5	Data 3	we/wy	Data EPP
6	6	Data 4	we/wy	Data EPP
7	7	Data 5	we/wy	Data EPP
8	8	Data 6	we/wy	Data EPP
9	9	Data 7	we/wy	Data EPP
10	10	PeriphCLK	we	-
11	11	PeriphAck	we	-
12	12	nAckReverse	we	-
13	13	X-Flag	we	-
14	14	HostAck	wy	-
15	32	PeriphRequest	we	-
16	31	nReverseRequest	wy	-
17	36	1284 Active	wy	-
18...25	19...30	GND		

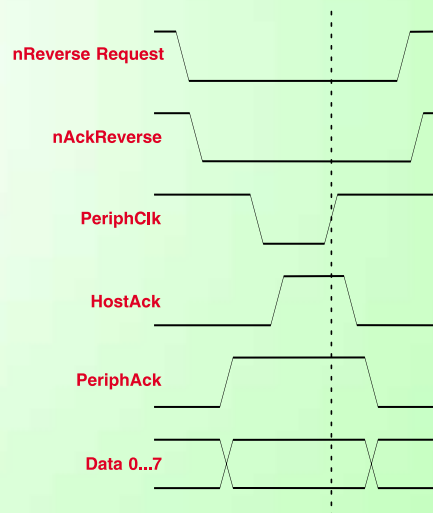
- podczas przesyłania danych z komputera do odbiornika zewnętrznego (*Forward Mode*) jest wejściem sygnału potwierdzającego przyjęcie danych przez odbiornik,
- podczas przesyłania danych z urządzenia zewnętrznego do komputera (*Reverse Mode*) sygnał ten przyjmuje poziom wysoki podczas przesyłania danych (*Data Cycle*), a niski podczas przesyłania polecenia (*Command Cycle*).
- *nAckReverse* - tą linią urządzenie zewnętrzne potwierdza przyjęcie żądania przełączenia się w tryb transmisji *Reverse Mode*.
- *X-Flag* - sygnał wykorzystywany podczas negocjowania rodzaju transmisji przez interfejs ECP. Ustawienie tego sygnału przez urządzenie zewnętrzne na poziom H oznacza, że obsługuje ono żądany przez komputer tryb pracy. W standardzie IEEE1284 opisano cztery tryby: *ECP*, *ECP Device ID*, *ECP with RLE compression* oraz *ECP Device ID with RLE compression*.
- *HostAck* - spełnia podwójną rolę:
 - podczas transmisji z komputera do urządzenia zewnętrznego (*Forward Mode*) logiczna jedynka na tej linii sygnalizuje transfer danych, natomiast logiczne zero informuje o przesyłaniu liniami danych polecenia (*Command Cycle*),
 - podczas transmisji z urządzenia zewnętrznego do komputera spełnia rolę linii potwierdzającej odbiór danych przez komputer.

- *PeriphRequest* - ustawienie przez urządzenie zewnętrzne tej linii w stan logicznego zera sygnalizuje żądanie odczytania przez komputer danych wystawionych przez to urządzenie (żądanie obsługi).
 - *nReverseRequest* - żądanie zmiany kierunku transmisji na urządzenie zewnętrzne->komputer. Jest on ustawiany przez komputer.
 - *1284 Active* - wyjście sygnalizujące logiczną jedynką pracę interfejsu w trybie ECP.
- Przez 8-bitowy port danych przesyłane są w obydwu kierunkach dane oraz polecenia, wśród których są rozróżniane: informacje o adresie urządzenia docelowego oraz polecenia związane z kompresją przesyłanych danych. Rodzaj informacji przesyłanych liniami danych (w trybie *Command Cycle*) jest sygnalizowany stanem logicznym linii *Data 7* (dla „1” liniami *Data 0..6* przesyłany jest adres urządzenia, a dla „0” liniami *Data 0..6* przesyłane jest słowo informujące o liczbie powtarzających się w pakiecie identycznych znaków). Taki sposób przesyłania danych powoduje, że długość słowa przesyłanego przez port danych zmniejsza się do 7 bitów.

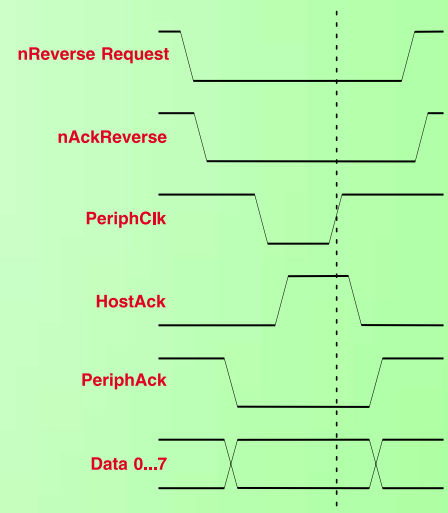
Zanim omówimy przebiegi ilustrujące role poszczególnych sygnałów interfejsu w różnych trybach pracy, przedstawimy graf przejść (zmian) stanów interfejsu ECP. Przedstawiono go na rys. 10. Jak widać, przejścia pomiędzy możliwymi stanami zostały szczegółowo zdefiniowane w opisie standardu ECP. Zmiana kierunku przesyłania danych z *Forward* na *Reverse* i odwrotnie jest możliwa w każdej chwili, o czym decydują odpowiednie kombinacje stanów logicznych na wejściach sterujących kierunkiem przepływu danych. Interfejs domyślnie jest uruchamiany w trybie *Compatible Mode*, w którym to zachowuje się jak klasyczny SPP. W trybie *Negotiate* jest ustalany obsługiwany przez urządzenie zewnętrzne tryb przesyłania danych (*ECP*, *ECP Device ID*, *ECP with RLE compression* lub *ECP Device ID with RLE compression*), natomiast *Setup* służy tylko do zapewnienia stabilizacji poziomów sygnałów z buforów wyjściowych (np. po przełączeniu z trybu *Open Collector* do *Push-Pull*).

Handshaking w ECP

Na rys. 11 pokazano sekwencję zmian stanów sygnałów podczas przesyłania danych z komputera do urządzenia współpracującego (*Forward Data Cycle*). Cykl zaczyna się od wystawienia przez komputer da-



Rys. 13.



Rys. 14.

Tab. 6. Mapa rejestrów interfejsu równoległego pracującego w trybie ECP.

Adres	Nazwa rejestru	Kierunek
Bazowy + 0	Data Port (w trybie SPP)	Zapis
	ECP Address FIFO (w trybie ECP)	Zapis/Odczyt
Bazowy + 1	Status Port	Zapis/Odczyt
Bazowy + 2	Control Port	Zapis/Odczyt
Bazowy + 400h	Data FIFO (w trybie ECP)	Zapis/Odczyt
	Data FIFO (w trybie <i>Parallel Port FIFO</i>)	
	Test FIFO (w trybie <i>Test</i>)	
	Configuration Register A (w trybie <i>Configuration</i>)	
Bazowy + 401h	Configuration Register B (w trybie <i>Configuration</i>)	Zapis/Odczyt
Bazowy + 402h	Extended Control Register	Zapis/Odczyt

nych na linii *Data 0...7*. Jak wcześniej wspomniano, przesyłanie danych wymaga ustawienia linii *HostAck* na „1”. Następnie urządzenie zewnętrzne jest informowane sygnałem *HostClk=0* o wystawieniu ważnych danych. Narastające zbocze sygnału *HostClk* powoduje zatrzaśnięcie danych w rejestrze wejściowym urządzenia zewnętrznego. W odpowiedzi na zmianę sygnału *HostClk* z logicznej „1” na „0”, urządzenie zewnętrzne potwierdza gotowość odebrania danych, co jest sygnalizowane zmianą stanu logicznego *PeriphAck* z „0” na „1”. Podobnie przebiega przesyłanie polecenia z komputera do urządzenia zewnętrznego (*Forward Command Cycle*), z tą różnicą (rys. 12), że sygnał *HostAck* przyjmuje stan logicznego „0”.

Nieco inaczej przebiega przesyłanie danych z urządzenia zewnętrznego do komputera (*Reverse Mode*). Na rys. 13 są widoczne przebiegi występujące podczas przesyłania danych w trybie *Reverse Mode*. Transfer rozpoczyna komputer zerując sygnał *nReverseRequest*, co wymusza na urządzeniu współpracującym potwierdzenie (lub jego brak w przypadku braku możliwości obsługi żądania) zmianą sygnału *nAckReverse* na logiczne „0”. W kolejnym kroku, po potwierdzeniu przez współpracujące urządzenie możliwości transferu danych do komputera, wystawia ono na linii *Data 0...7* dane, co jest sygnalizowane zmianą stanu linii *PeriphAck* z „0” na „1” (w przypadku

przesyłania polecenia - *Command Cycle* - sygnał ten przyjmuje stan logicznej „1” - rys. 14). Gotowość wystawionych danych urządzenie zewnętrzne sygnalizuje zmianą stanu logicznego na linii *PeriphClk* z „1” na „0”. Komputer potwierdza gotowość do odbioru danych zmieniając stan logiczny linii *HostAck* z „0” na „1”. Następnie, narastające zbocze sygnału *PeriphClk* wyznacza moment wpisania przesyłanych danych do rejestru wejściowego portu komputera. Odbiór danych przez komputer jest potwierdzany zmianą stanu logicznego na linii *HostAck* z „1” na „0”.

Pomimo tego, że liczba „kroków” niezbędnych do transferu danych w trybie ECP jest znacznie większa od liczby kroków koniecznych do transferu danych w trybie SPP (zaledwie 5), szybkość pracy ECP jest większa, a to dlatego, że ta procedura jest realizowana sprzętowo.

Rejestry

Ze względu na odmienne mechanizmy wykorzystywane przez interfejs pracujący w trybie ECP, zupełnie inną strukturę niż dla trybu SPP i EPP mają rejestry interfejsu. W tab. 6 podano mapę adresów rejestrów niezbędnych do poprawnej pracy portu ECP. Dostęp do rejestrów konfiguracyjnych A i B jest możliwy tylko w trybie konfiguracji interfejsu. Rejestry konfiguracyjne A i B służą do ustawiania parametrów pracy interfejsu „od strony” złącza ISA (ustawianie przerwania, uaktywnianie kanałów DMA, spo-

Tab. 7. Funkcje bitów rejestru ECR (base + 402h).

Bity	Funkcja
7...5	Wybór trybu pracy interfejsu: 000 - Standard Mode 001 - Byte Mode 010 - Parallel Port FIFO Mode 011 - ECP FIFO Mode 100 - EPP Mode 101 - zarezerwowane 110 - FIFO Test Mode 111 - Configuration Mode
4	ECP Interrupt
3	DMA Enable
2	ECP Service
1	FIFO Full
0	FIFO Empty

sób pracy FIFO itp.), natomiast rejestr ECR (tab. 7) odpowiada za:

- ustawienie trybu pracy interfejsu (w tym przełączenie go w tryb konfiguracji),
- uaktywnienie obsługi DMA i przerwań,
- kontrolę stanu bufora FIFO (pusty/pełny).

Szczegółowy opis funkcji tych rejestrów pominiemy, ponieważ jest on ściśle związany z oprogramowaniem komputera sterującego pracą interfejsu. Opis przedstawiony w artykule umożliwi natomiast samodzielną realizację urządzenia wyposażonego w interfejs zgodny z ECP wykonany programowo lub sprzętowo (np. w układzie PLD). Czytelników zainteresowanych opisem funkcji rejestrów zachęcam do odwiedzenia naszego działu *Download* (www.ep.com.pl) lub sięgnięciu po płytę CD-EP02/2002B, na której publikujemy wybrane materiały źródłowe związane z tematyką interfejsów równoległych SPP/EPP i ECP.

Tomasz Jakubik, AVT

Dodatkowe informacje

Dodatkowe informacje można znaleźć w Internecie pod adresami:

- <http://www.beyondlogic.org/pardebug/pdebug.htm>,
- <http://www.beyondlogic.org/spp/parallel.pdf>,
- <http://www.beyondlogic.org/epp/epp.pdf>,
- <http://www.beyondlogic.org/ecp/ecp.pdf>,
- <http://www.lvr.com/parport.htm>,
- <http://www.lpt.com/Downloads/downloads.htm>.