Cyfrowa archeologia – addendum, czyli mikrokomputer dino-85

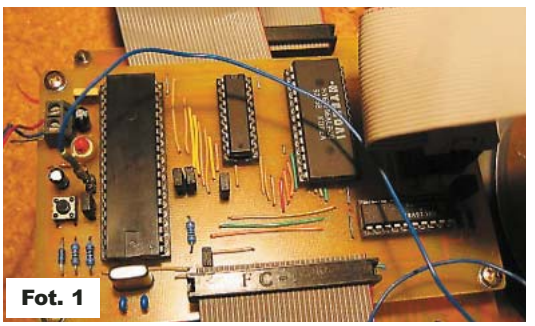
Poniższy tekst może wydać się odrobinę egzotyczny (szczególnie dla młodszych Czytelników), ale równie nietypowa jest jego geneza. Podczas prac nad artykułem „Cyfrowa archeologia...” (EdW 7/2008) urządziłam sobie wycieczkę w świat mikroprocesorów z końca lat siedemdziesiątych, kostek jakże pociesznie wyglądających w konfrontacji ze współczesnymi, „wszystkomającymi” kontrolerami. Kostek czasem kłopotliwych w wykorzystaniu, układów o ubogiej dokumentacji, ale z drugiej strony nadających urządzeniu, które je wykorzystuje, pewien specjalny, nostalgiczny klimat. Możliwe, że jestem niepoprawną romantyczką, nawet jeżeli chodzi o elektronikę, ale chyba dlatego tak urzekła mnie strona internetowa Izabeli Malcolm (http://izabella.sbprojects.com), na której można znaleźć wiele projektów opartych o stare, ośmiobitowe mikroprocesory. Z projektów tych najbardziej spodobał mi się SITCOM (San & Izabella Training Computer), mały komputerek edukacyjny zaprojektowany przez Izabelę Malcolm i Sana Bergmansa, a wykorzystujący mikroprocesor Intel 8085. Pomyślałam, że to byłoby całkiem ciekawe mieć podobną zabawkę na swoim biurku. Jednak nie chciałam powielać gotowego projektu. I tak rozpoczęłam własne prace i poszukiwania, aby podobny komputerek sobie zbudować. Niestety, w pewnym momencie zabrakło motywacji i skończyło się na mniej lub bardziej zaawansowanych eksperymentach sprzętowych i strzępkach kodu w assemblerze. Przełom nastąpił po zakończeniu prac nad „Cyfrową archeologią...”. Nagle odkryłam, że buszowanie w sieci za dokumentacją starych układów mikroprocesorowych, lektura książek z lat 70.–80. traktujących o programowaniu mikroprocesorów, to w sumie doskonała zabawa. A biorąc pod uwagę fakt, że byłam w posiadaniu większości układów scalonych, o których właśnie czytałam – coraz częściej wracała myśl, aby wskrzesić porzucony niegdyś pomysł na własny, staroświecki komputerek. I tak właśnie powstał dino-85.

Dlaczego dino-85?

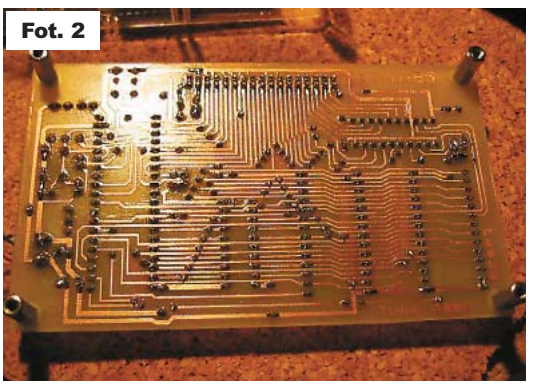
A dlatego, że system zbudowany jest na bazie procesora 8085, a to przecież taki mikroprocesorowy dinozaur. No i chyba sympatycznie brzmi. Wprawdzie przez chwilę rozważałam wykorzystanie kostki Intel 8080 (co byłoby jeszcze większym krokiem wstecz), ale po namyśle zrezygnowałam. Ten procesor wymaga kilku dodatkowych układów wspomagających, zasilany jest napięciami o trzech różnych wartościach, czyli same kłopoty. Układ 8085 tych cech (wad?) nie ma, multipleksowana magistrala danych i adresowa też nie jest problemem – więc ostatecznie stanęło na 8085. W pierwotnym zamyśle system miał składać się z dwóch modułów: płytki z procesorem, układami pamięci i jakimś prostym podsystemem wejścia/wyjścia oraz dodatkowej płytki tylko i wyłącznie z układami we/wy, ale już bardziej zaawansowanymi. Ta koncepcja upadła na rzecz zupełnie innej i jak się okazało – o wiele lepszej, o której będzie dalej.

Moduł procesora

Podstawą dino-85 jest mała płytka z procesorem 8085A (U1), pamięciami (U3,U4) i dekoderem adresowym (U5). Większość użytecznych sygnałów wyprowadzona jest na złącze igłowe (Z2) w formie magistrali zewnętrznej. Dekoder adresowy U5 zrealizowany na układzie GAL16V8 definiuje sprzętową mapę pamięci oraz przestrzeni wejścia/wyjścia, którą można podsumować następująco: 8kB pamięci (E)EPROM, 32kB statycznej pamięci RAM, sześć segmentów I/O każdy o pojemności 16 adresów. Generator sygnału zegarowego wbudowany w procesor pracuje z kwarcem 6.144 MHz (Q1), co daje częstotliwość taktowania procesora równą 3,072 MHz.



Fot. 1

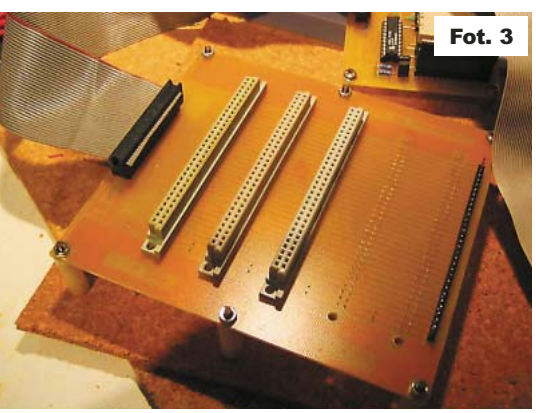


Fot. 2

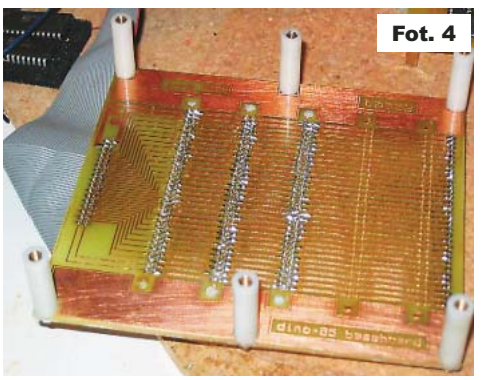
Nie jest to zbyt wiele jak na dzisiejsze standardy, ale powinno wystarczyć do większości niekrytycznych czasowo zastosowań. Schemat ideowy tego modułu przedstawia **rysunek 1**, a **fotografie 1, 2** gotową płytkę od strony elementów i druku.

Płytka bazowa

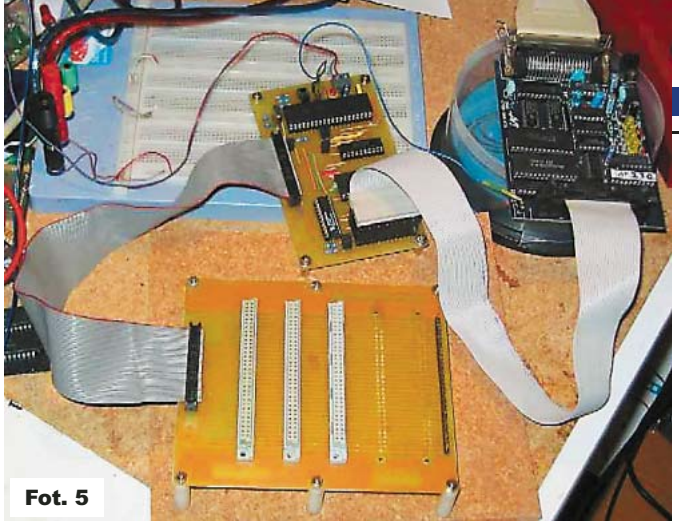
Drugi element dino-85 to płytka bazowa dla modułów rozszerzających – **fotografie 3, 4**. To spory kawałek laminatu, na którym umieściłam przemysłowe złącza DIN typu 41.612 (takie złącza produkowano kiedyś w Polsce w zakładach Eltra). Na razie, wlutowałam tylko trzy złącza, na dwa pozostałe jest na płytce miejsce, dodatkowo płytka zawiera jednorzędowe złącze igłowe, co znakomicie ułatwia mi przyłączanie modułu do płytek typu solderless-breadboard. Wspomniane złącza DIN są o tyle sympatycznym wynalazkiem, że bardzo łatwo można zakupić uniwersalne płytki prototypowe w formacie eurokarty, dostosowane właśnie do złączy typu DIN. Płytka bazowa dołączona jest do płytki procesora za pomocą kawałka 34-żyłowej taśmy sygnałowej, takiej jakiej używa się do podłączania stacji dysków do płyty głównej w komputerach PC. Na **fotografii 5** widać cały zestaw wraz z przyłączonym emulatorem pamięci EPROM.



Fot. 3



Fot. 4



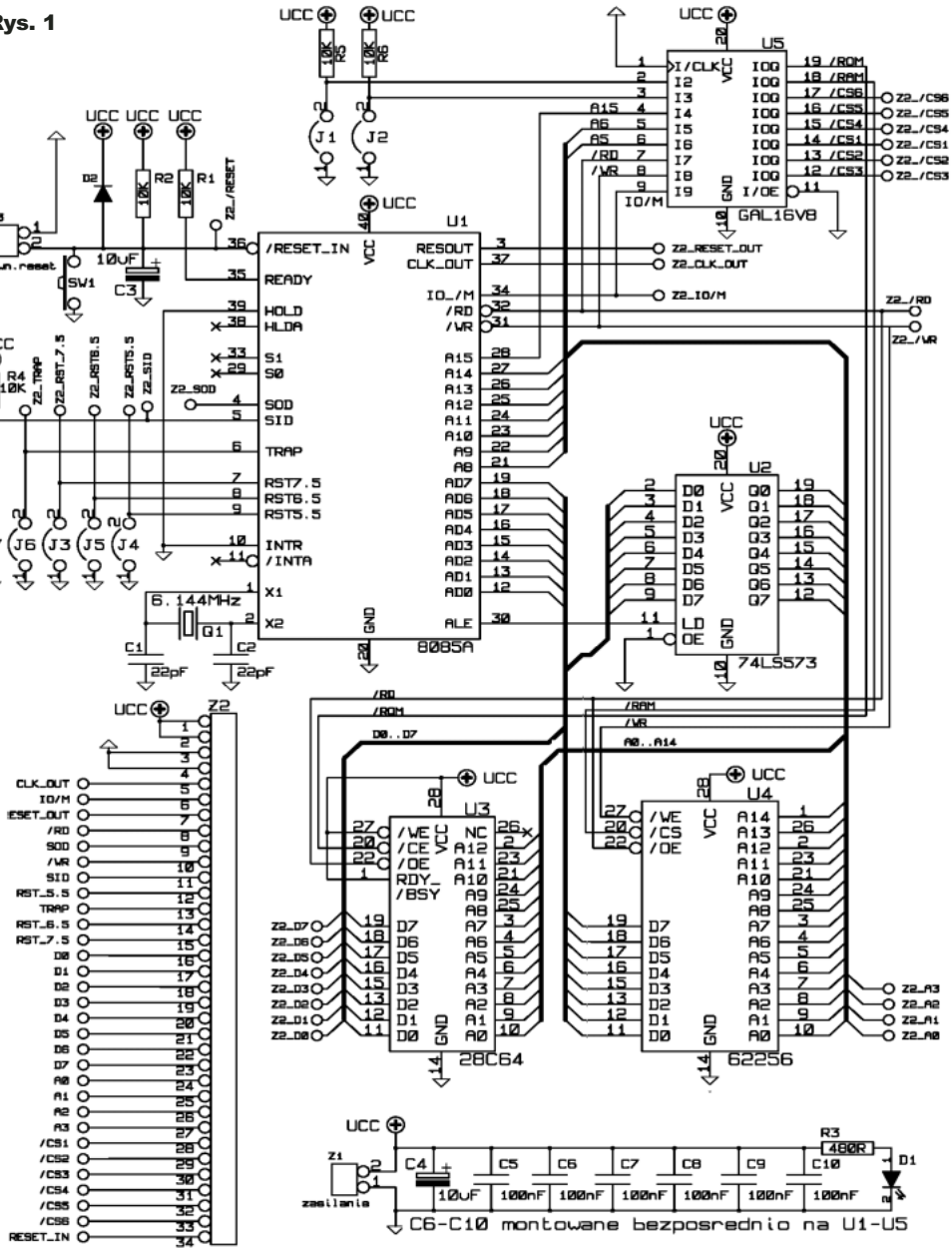
Fot. 5

Moduł komunikacyjny

Trzeci element systemu to moduł komunikacyjny w formie karty dla płytki bazowej. Na wstępie przyznam się do jednej rzeczy... pierwsza realizacja tej karty była nieco inna, bogatsza. Zawierała dwa układy UART oraz możliwość podłączenia sprzętowego mostka RS232/1-Wire, czyli kostki DS2480B. Niestety, na skutek nieuwagi podczas eksperymentów, karta uległa poważnemu uszkodzeniu i z tego powodu w tym tekście prezentuję efekt prac nad drugą wersją modułu. Jest to nieco skromniejszy układ, ale myślę, że do zabaw z komputerkiem w zupełności wystarczy. Jak pokazuje schemat na **rysunku 2** – moduł komunikacyjny jest raczej średnio skomplikowaną konstrukcją. Zawiera układ UART typu 8251A oraz po jednym ośmiobitowym porcie wejściowym i wyjściowym. UART (U3) służy do komunikacji z komputerem nadrzędnym za pomocą RS232, dlatego też wyposażony jest w konwerter poziomów napięć na układzie MAX232 (U4). Dodatkowo, wolne linie UART-a, przeznaczone do sterowania pracą modemu (DRT, RTS, DSR), posłużyły do realizacji interfejsu I^{2}C. A skoro I^{2}C już jest, to niejako przy okazji dołożyłam zegar czasu rzeczywistego PCF8583 (U7). Przydziałem adresów I/O do poszczególnych układów zajmuje się lokalny dekoder adresowy (U1), podrzędny względem tego z płytki procesora, a zrealizowany także na układzie GAL16V8. Port wyjściowy (U5) to ośmiobitowy rejestr D typu 74LS574, można do niego podłączyć diody LED w celu sygnalizacji, można też podłączyć standardowy moduł LCD (oprogramowanie dino-85 wspiera takie podłączenie zestawem prostych procedurek). Ośmiobitowy port wejściowy na bazie układu 74LS541 (U6) obsługuje prostą klawiaturę, choć można go wykorzystać dowolnie inaczej, na przykład podłączyć przetwornik A/C. Odnośnie do komunikacji via RS232 – transmisja z PC jest asynchroniczna, z wykorzystaniem tylko i wyłącznie linii TxD oraz RxD, a wystawiany przez PC sygnał DTR służy do zdalnego resetowania komputerka. Oprogramowanie systemowe dino-85 nie wykorzystuje przerwań do obsługi transmisji szeregowej, niemniej jednak możliwość taka istnieje, ponieważ zworka J1 opcjonalnie dołącza UART do wejścia przerwania maskowalnego RST_5.5 procesora 8085. **Fotografia 6** przedstawia prototyp opisywanego modułu w trakcie uruchamiania na płytce stykowej. Ten etap prac ma to do siebie, że aktualnie działający układ momentami delikatnie odbiega od wstępnie założonego schematu ideowego. Po prostu ciągle szukam optymalnych rozwiązań i jest spora szansa, że finalny schemat będzie nieco inny. I stąd też chwilowy brak nowej płytki drukowanej.

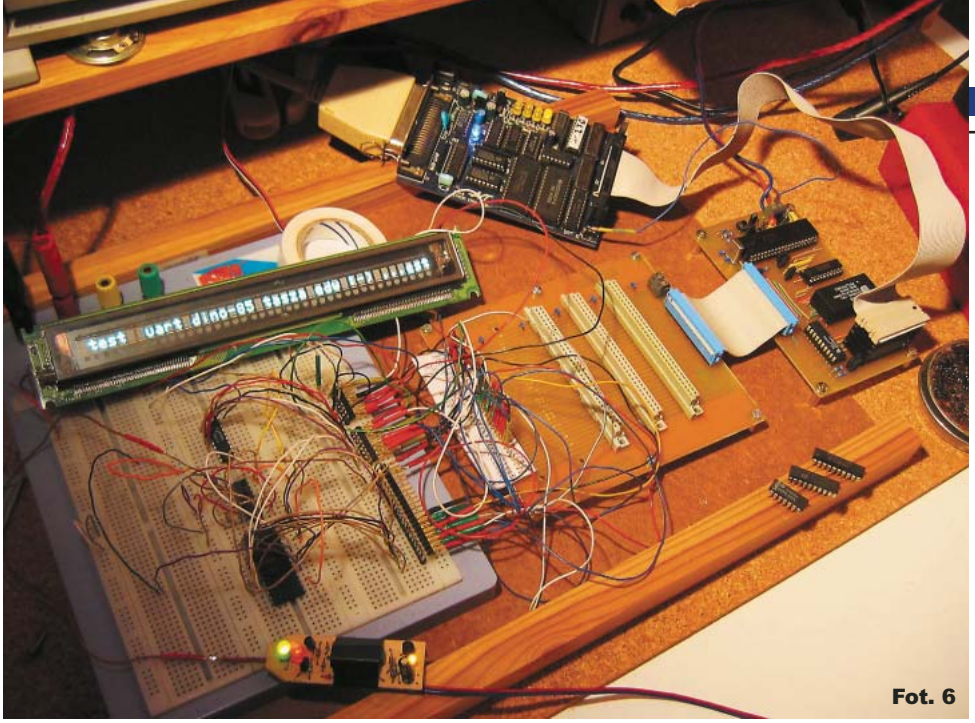
W formie nazwijmy to – zupełnie teoretycznego projektu – są jeszcze dwa inne moduły: cyfrowego I/O wraz z układami czasowo-licznikowymi (na układach 8255 i 8254) oraz drugi, z przetwornikami A/C i C/A. Z ich fizyczną realizacją na razie się wstrzymałam, ponieważ chcę ostatecznie doszlifować oprogramowanie systemowe.

Rys. 1



Dino-mon

Cały opisany powyżej sprzęt byłby tylko elektronicznym złomem bez odpowiednio przygotowanego oprogramowania – to chyba oczywiste. Dlatego też teraz kilka zdań o oprogramowaniu pokładowym. Nazwałam je „dino-mon”, choć skrót od monitor (kodu maszynowego) to tak troszkę na wyrost. Dino-mon to raczej swego rodzaju loader pozwalający wgrać do dino-85 przygotowane na PC oprogramowanie w formie rekordów Intel HEX oraz je uruchomić. Dodatkowo, dino-mon umożliwia wykonanie prostych operacji typu zapis/odczyt na dostępnych w systemie portach wejścia/wyjścia oraz zainstalowanej pamięci, tego typu poleceń dino-mon obsługuje kilkanaście. Reszta oprogramowania to zestaw procedur systemowych do wykorzystania w aplikacjach użytkowych, które zapewniają obsługę komunikacji szeregowej, magistrali I^{2}C (i zapiętego na niej RTC) oraz modułu LCD i prostej klawiaturki. Dino-mon pracuje w sposób konwersacyjny – oczekuje od użytkownika poleceń (komend) a status ich wykonania (lub wyniki pracy) odsyła na terminal, komunikacja odbywa się za pomocą łącza



Fot. 6

RS232. Dino-mon ma jeszcze jedną cechę: gdy jako pamięć RAM wstawimy do dino-85 kostkę NV-RAM (np. DS1230), to istnieje możliwość automatycznego uruchomienia załadowanej aplikacji użytkownika zaraz po włączeniu zasilania, bez żadnej ingerencji z zewnątrz.

...zamiast podsumowania

Pewnie niejeden Czytelnik zacznie się zastanawiać – po co w ogóle zawracać sobie głowę starym, archaicznym mikroprocesorem, gdy mamy w zasięgu ręki procesory i kontrolery o parametrach lepszych o rząd wielkości. No cóż, to zależy, jak rozumie się słowo „hobby". Dla mnie prace przy dino-85 to fantastyczna przygoda, a satysfakcja z uruchamiania coraz nowszych elementów konstrukcji w zupełności rekompensuje wiele zarwanych nocy. Poza tym, to chyba dość niezwykłe, pobawić się samodzielnie wykonanym i oprogramowanym systemem mikroprocesorowym bazującym na kostce, która ma prawie trzydzieści lat. Do czego bardziej praktycznego niż zabawa z assemblerem 8085 taki dino-85 może posłużyć? Zastosowań może być wiele: proste systemy akwizycji danych, centralki alarmowe, programatory (E)PROM/FLASH, testery układów scalonych, sterowanie kolejką PIKO czy robotem... Dino-85 ma konstrukcję modułową i raczej dość elastyczną, więc łatwo go zaadaptować do wielu celów, traktując jako jednostkę sterującą większą całością. Jeżeli ktoś pokusi się o powielenie lub zmodyfikowanie tego projektu – miła wiadomość: wszystkie układy (włącznie z procesorem 8085A) można kupić w popularnych sklepach internetowych. Wprawdzie szanse na zakup oryginalnych kostek Intela są praktycznie zerowe, ale układy w wersji CMOS na przykład firmy NEC czy OKI spokojnie dostaniemy. Tylko pamiętajmy, że to nie będzie tania zabawa. Znacznie lepiej potrzebne elementy pozyskać ze starego złomu komputerowego.

Na początku tego tekstu wspominałam o książkach. Znakomitą większość potrzebnych mi informacji znalazłam w następujących publikacjach: „Układy mikroprocesorowe 8080/8085 w modułowych systemach sterowania" (J. Pieńkos, S. Moszczyński, A. Pluta, WKŁ, Warszawa 1988), „Podstawy i praktyka programowania mikroprocesorów" (J. Grabowski, S.Koślacz, WNT, Warszawa 1980) oraz „8080-8085 Assembly Language Programming" (Intel, 1978). W polskiej prasie elektronicznej znalazłam tylko jeden (za to dwuczęściowy) artykuł traktujący o konstrukcji opartej na Intel 8085. To bardzo ciekawy tekst autorstwa pana Leszka Leśniewskiego, zatytułowany „Prosty uniwersalny system mikroprocesorowy", a opublikowany w „Radioelektroniku", w numerach 11/86 oraz 12/86. Za zgodą Redakcji „Re" (dziękuję!) materiał ten w formie elektronicznej dostępny jest także na naszym Elportalu.

Dino-85 powstał głównie z wykorzystaniem darmowego oprogramowania. Część elektroniczną projektu, czyli schematy i pcb, wykonałam w KiCAD. Oprogramowanie systemowe oraz programy testowe pisałam, korzystając z SB-Assembler autorstwa Sana Bergmansa. Moją wątpliwość budzi jedynie status kompilatora dla PLD czyli programu eqn2jed firmy National Semiconductor, którym przygotowałam wsady do układów GAL. To bardzo stary kompilator i można go odnaleźć na wielu uczelnianych serwerach ftp, czy legalnie? Tego nie wiem. Zawsze można się posiłkować innym narzędziem, choćby pakietem WinCUPL udostępnionym przez firmę Atmel, ostatecznie dekodery adresowe wykonać na układach TTL – wtedy to już będzie prawdziwy dinozaur...

Natasza Nordecka
bienata@wp.pl
strona domowa projektu:
http://tasza.republika.pl/dino85

Rys. 2