

# GALER - programator GAL-i do Amigi, część 1

## AVT-5049



PROJEKT  
Z OKŁADKI

*W artykule przedstawiamy prawdziwy hit - programator układów GAL, do którego jest dostępne shareware'owe oprogramowanie w wersji na Amigę. Prezentowany w artykule układ jest wzorowany na jednym z wielu dostępnych w Internecie.*

Układy programowalne typu GAL są - mimo swojej długiej historii - nadal najczęściej stosowane w aplikacjach nie wymagających stosowania układów o dużej skali integracji. Doskonale nadają się do budowania dekodерów adresowych w systemach mikroprocesorowych i na kartach komputerowych, dzięki temu, że można nimi zastąpić kilka standardowych układów cyfrowych. Ich stosowanie jest uzasadnione w urządzeniach o niewielkich wymiarach. Za ich pomocą można także zabezpieczać swoje opracowanie, wbudowując mniej lub bardziej istotny fragment „logiki” układu w zabezpieczony przed odczytem układ programowalny. Nie są to oczywiście wszystkie powody, dla których GAL-e cieszą się nieustającym powodzeniem, niemniej jednak wystarczające do tego, żeby przedstawić ten prosty w wykonaniu programator.

### Opis układu

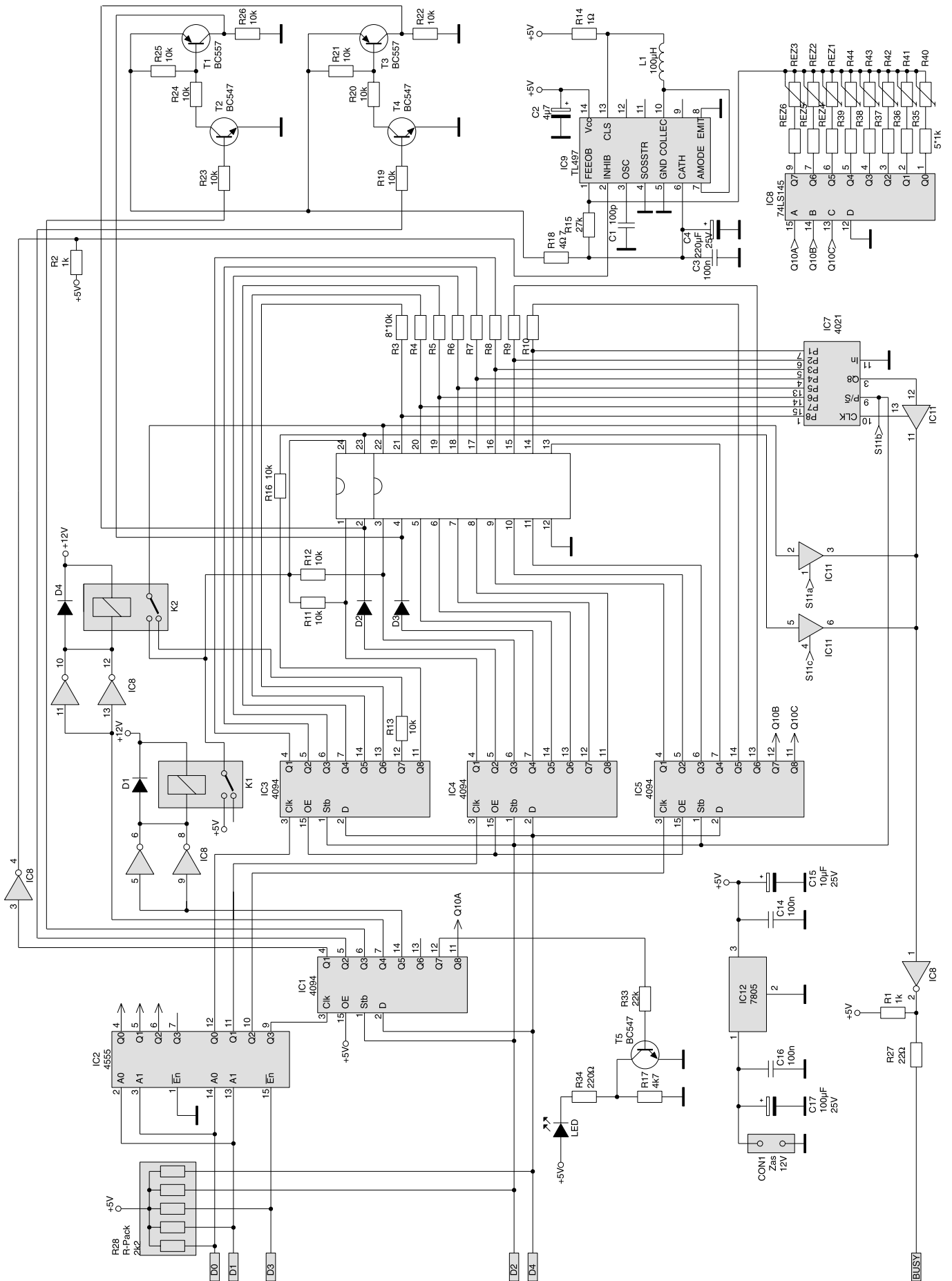
Układ programatora, którego schemat pokazano na **rys. 1**, jest nieco zmodyfikowaną wersją programatora opracowanego przez Christiana Habermanna. Zmiany wprowadzone w stosunku do oryginału są następujące:

- Dodałem rezystor R17, który powoduje lekkie świecenie diody LED w czasie, kiedy miała być wyłączona. Dzięki temu wiadać, czy urządzenie jest włączone do sieci.
- Zmodyfikowałem układ zasilania kontaktronów K1, K2 tak, aby możliwy był wybór napięcia zasilania ich cewek pomiędzy 5 a 12V (kontaktrony na 12V w obudowie DIL są łatwiej osiągalne).
- Dodałem stabilizator 7805, dzięki czemu programator można zasilić z dowolnego źródła napięcia o wartości 8...15V.
- Zmodyfikowałem układ połączeń do podstawki zatrzaskowej tak, że umieszczona jest ona od strony (umownej) druku. Ułatwia to montaż płytki programatora w obudowie.

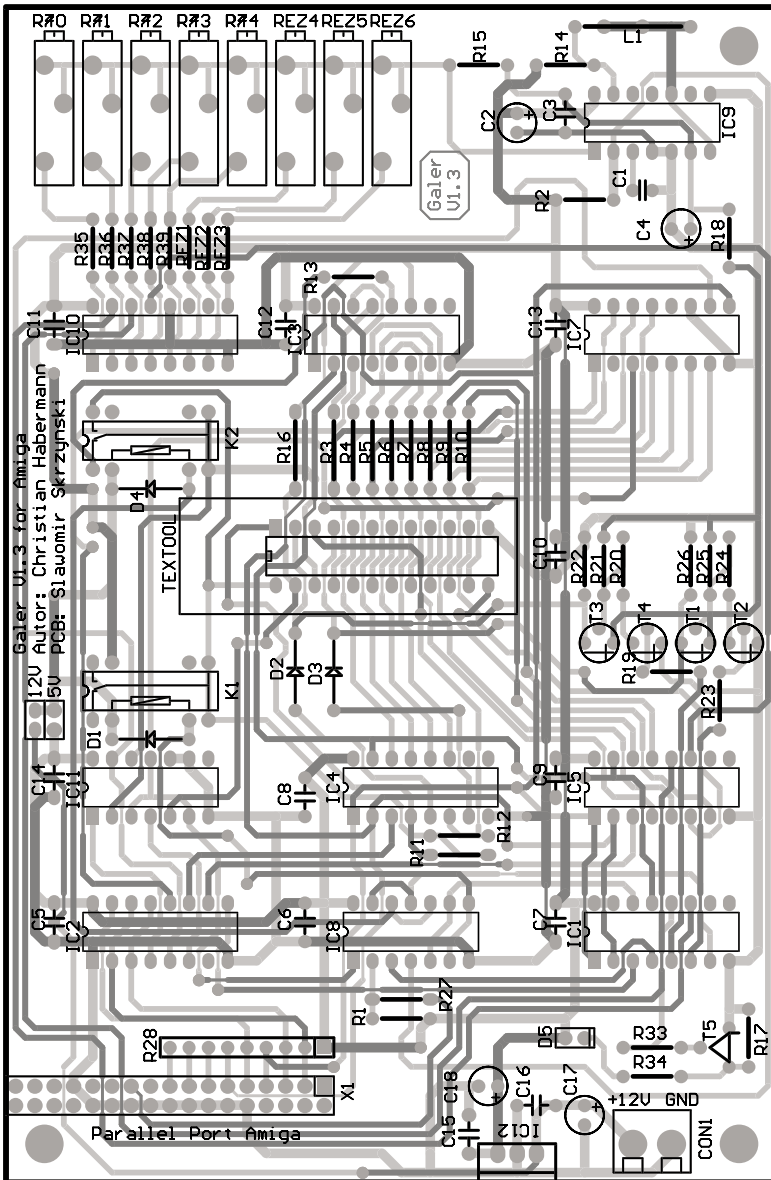
Działanie układu przebiega następująco. Za pomocą sygnałów D0, D1, D3, podawanych na wejścia A0, A1, En układu IC2, wybieramy rejestr (IC1, IC3, IC4 lub IC5), do którego będą szeregowo wpisywane dane oraz z którego wyprowadzenia programowanego układu będziemy czytać dane (IC7, IC11). Wysyłany bit pojawia się na linii D4, a sygnał zegarowy jest podawany na linię

### Charakterystyka i możliwości programatora z oprogramowaniem:

- ✓ programuje, kopiuje, odczytuje, weryfikuje układy: GAL16V8, GAL20V8, GAL22V10, GAL20RA10,
- ✓ współpracuje z wszystkimi komputerami Amiga,
- ✓ jest dołączany do złącza portu Parallel,
- ✓ umożliwia ustawienie bitów zabezpieczających,
- ✓ wbudowany kompilator,
- ✓ wbudowany deassembler,
- ✓ testowanie zaprogramowanych układów.



Rys. 1. Schemat elektryczny programatora.



Rys. 2. Rozmieszczenie elementów na płycie drukowanej.

bitu D3 (aktywne jest ujemne zbocze tego sygnału). W trakcie zapisu do rejestru 4094 dane na jego wyjściu nie zmieniają się, a to za sprawą wewnętrznego zatrasku wyjściowego. Po ośmiu impulsach zegarowych dane zostają przepisane do zatrasku wyjściowego narastającym zboczem sygnału z linii D2.

Bity odczytane podczas odczytu są podawane na linię BUSY portu równoległego. Dane mogą pochodzić z wyprowadzenia 23 programowanego układu (bramka IC11c), wyprowadzenia 22 (bramka IC11a) lub rejestru IC7. Dane do IC7 wpisywane są impulsem na linii D2 portu równoległego. Podanie adresu „01“ na D0, D1 powoduje pojawienie się sygnału

strobującego na wyjściu IC2 oznaczonym S11b. W wyniku tego dane w rejestrze IC7 są przesuwane, a ponadto jest uaktywniana bramka trójstanowa IC11b. Dzięki temu dane można odczytać na linii BUSY portu równoległego. Inwerty układu IC8 sterują przekaźnikami kontaktronowymi załączającymi napięcie zasilania programowanego układu (K1) oraz podającymi napięcie zasilania programowanego układu na wyprowadzenie odpowiednie dla układów w obudowach 20 i 24-wyprowadzeniowych.

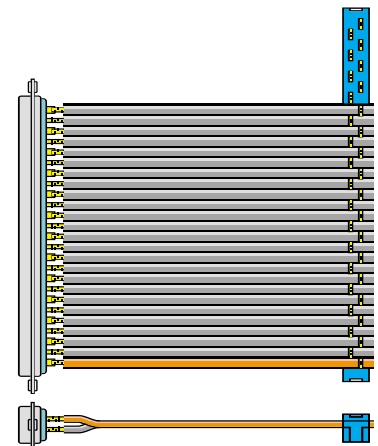
Tranzystory T1...T4 załączają napięcie programujące uzyskiwane z przetwornicy impulsowej IC9 typu TL497. Pojawienie się wysokiego poziomu napięcia na wy-

prowadzeniu 2 przetwornicy blokuje jej pracę. Napięcie programujące wybierane jest za pośrednictwem dekodera IC10, do którego wyjść dołączono potencjometry, za pomocą których precyzyjnie ustala się wartość napięcia programującego.

### Montaż i uruchomienie

Sposób zamontowania podstawki i diody LED oraz sposób umieszczenia programowanego układu w podstawce pokazano na rys. 2. Elementy montujemy w tradycyjny sposób, począwszy od najmniejszych rozmiarami do największych. Pod wszystkie układy polecam użycie podstawek. Diode LED i podstawkę zatraskową montujemy od strony lutownia (umownej). Jeśli programator będziemy wykorzystywać sporadycznie, stosunkowo drogą podstawkę ZIF można zastąpić podstawką tulipanową. Zależnie od napięcia zasilania programatora, dla IC12 konieczne może być zastosowanie radiatora. Na kawałku taśmy FLAT34, z której odcinamy 9 żył, zaciskamy złącze IDC34 oraz wtyk DB25-M. Jeśli nie posiadamy wtyku DB25 zaciskanego na taśmie, przewody można przylutować. Szczegóły montażu pokazano na rys. 3. Wtyk DB25 będzie zamocowany do obudowy, dlatego do połączenia programatora z komputerem wykorzystujemy typowy przedłużacz 25-przewodowy.

Bez podłączenia programatora do komputera i z pustymi podstawkami pod układy scalone podłączamy do złącza ARK napięcie zasilające i sprawdzamy obecność napięcia +5V na wyprowadze-



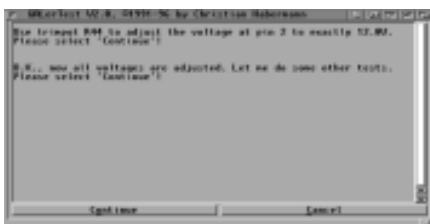
Rys. 3. Sposób wykonania kabla połączeniowego.



Rys. 4. Widok okna wyboru wersji programatora.

niach wszystkich układów scalonych. Jeśli wszystko jest dobrze, odłączamy zasilanie, umieszczamy wszystkie układy w podstawkach, łączymy programator z komputerem i po ponownym włączeniu zasilania uruchamiamy program *GalerTest*. Po tekście ostrzegawczym pojawi się okno wyboru wersji programatora, wybieramy V1.3 (rys. 4). Następnie wybieramy kolejno testy sprzętu.

*Pierwszy test* - ustawia na wszystkich wyprowadzeniach podstawki ZIF stan logiczny 0. Do testowania poziomów napięcia najlepiej użyć sondy logicznej, ponieważ na mierniku nie odróżnimy poziomu 0 od braku napięcia. W obu przypadkach miernik wskaże 0V. Jeśli test wypadł niepomyślnie to mamy spory kłopot. W pierwszej kolejności należy sprawdzić IC2 i IC1. Na wyprowadzeniu 13 IC1 powinien być poziom H. Jego brak świadczy o złym sterowaniu przez IC2 lub o braku sygnału stroboju i danych (linie D2 i D4 portu równoległego). Aby zlokalizować przyczynę błędu podłączmy sondę logiczną do wyprowadzenia 1 IC1 i ponownie uruchamiamy program testujący. Na wyprowadzeniu tym powinny pojawić się dodatnie impulsy. Podobnie na wyprowadzeniach 2 i 3. Brak impulsów na wyprowadzeniu 1 świadczy o zwarciu ścieżek łączących wyprowadzenia 1 układów IC1, IC3, IC4, IC5 lub uszkodzeniu (przebiciu) któregoś z wymienionych



Rys. 5. Widok okna programu testującego.

układów. Aby to stwierdzić wystarczy usunąć IC1, IC3...IC5 z podstawek. Jeśli nadal brak jest impulsów, to zwarta jest ścieżka, a jeśli impulsy pojawiły się, to uszkodzony jest jeden z układów. Umieszczając kolejno układy znajdziemy ten, który blokuje linię. Podobnie postępujemy w przypadku braku impulsów na wyprowadzeniu 2 IC1. Jeśli brak impulsów na wyprowadzeniu 3 IC1, prawdopodobnie jest uszkodzony IC2 lub zwarta ścieżka doprowadzająca dane do niego. Pojawienie się impulsów na wyprowadzeniu 15 IC2, a brak impulsów na wyjściach (wyprowadzenia 9, 10, 11, 12) świadczy o uszkodzeniu IC2. Gdy usunęliśmy usterkę lub test wypadł pomyślnie naciskamy przycisk OK.

*Test 2* - Sprawdzenie T5 - dioda LED powinna zaświecić.

*Test 3* - Zgaszenie diody LED.

*Test 4* - Sprawdzenie działania kontaktronu K1 - na wyprowadzeniu 24 podstawki ZIF pojawia się napięcie +5V.

*Test 5* - Sprawdzenie działania kontaktronu K2 - na wyprowadzeniu 22 podstawki ZIF pojawia się napięcie +5V.

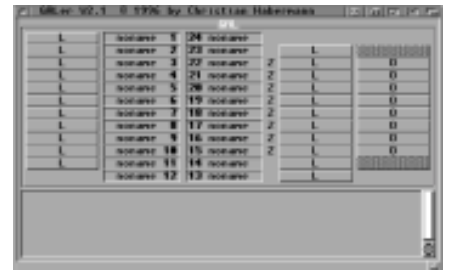
*Test 6* - Na wszystkich wyprowadzeniach podstawki ZIF (pomińjąc wyprowadzenie 12) pojawia się poziom logiczny H.

*Test 7* - Wyprowadzenia 1, 3, 5, 7, 9, 11, 14, 16, 18, 20, 22, 24 przyjmują poziom H, pozostałe L.

*Test 8* - Wyprowadzenia 2, 4, 6, 8, 10, 13, 15, 17, 19, 21, 23 przyjmują poziom H, pozostałe L.

Testy 6..8 umożliwiają sprawdzenie wystąpienia ewentualnych zwarć na ścieżkach doprowadzających napięcia z układów scalonych i kontaktronów do podstawki ZIF.

*Test 9* - Pozwala zweryfikować działanie przetwornicy IC9, dekodera IC10 oraz tranzystorów T3, T4. Na wyprowadzeniach 12, 13, 14, 15 IC10 powinny być logiczne zera, na wyjściu 1 napięcie rzędu 50...300mV. Napięcie większe niż 500mV świadczy o uszkodzeniu IC10. Na wyprowadzeniu 2 IC9 powinien być poziom H. Stan L oznacza uszkodzenie IC8. Na wyprowadzeniu 2 IC9 powinno wystąpić napięcie większe niż 5V (7...30V). Napięcie to regulujemy potencjometrem R40. Na kolekto-



Rys. 6. Widok głównego okna obsługującego programator.

rze T3 powinno pojawić się właśnie regulowane napięcie. Ustawiamy jego wartość na 16,50V, kontrolując na nóżce 2 podstawki ZIF.

*Test 10* - Ustawiamy napięcie 15,75V

*Test 11* - Ustawiamy napięcie 14,50V

*Test 12* - Ustawiamy napięcie 14,00V

*Test 13* - Ustawiamy napięcie 12,00V

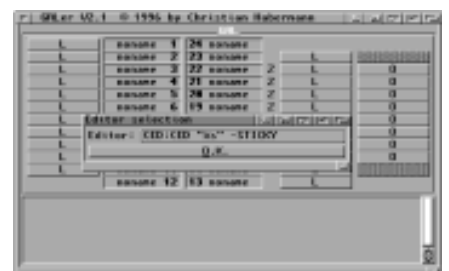
Po naciśnięciu OK pojawi się w oknie komunikat pokazany na rys. 5.

*Test 14* - Kontrola napięcia 12,00V na wyprowadzeniu 4 podstawki ZIF. W ten sposób zostały sprawdzone T1 i T2.

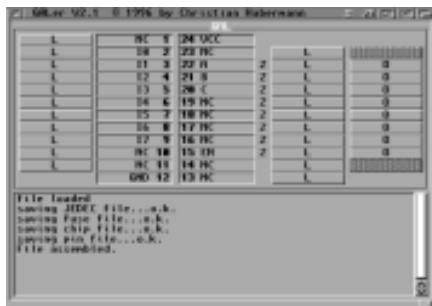
*Test 15* - Sprawdzenie IC7. Program wygeneruje dane na wyjściach rejestrów, po czym odczyta je przez IC7 i IC11. Wyniki testu są przedstawiane w oknie programu.

### Obsługa programatora

Po przetestowaniu programatora możemy uruchomić program sterujący jego pracą - *Galer*. Główne okno programu pokazano na rys. 6. Jeśli rozmiar wyświetlanego okna jest za mały, możemy go zmienić w standardowy sposób. W pierwszej kolejności wybieramy „V1.3” z podmenu *Hardware-Version* w menu *Project*. Następnie w menu *Tools* wybieramy podmenu *Select Editor*. W oknie dialogowym, które pojawi się, wpisujemy nazwę używanego edytora np. CED (rys. 7). W menu *GAL-*



Rys. 7. Okno wyboru edytora.



Rys. 8. Widok głównego okna programu sterującego po poprawnym zakończeniu kompilacji.

Type warto zaznaczyć opcję *Type-Requester*. Dzięki temu przed każdą operacją (programowanie, kopiowanie, sygnatura) pojawi się okno umożliwiające wybór typu GAL-a. Ustawienia zapisujemy wywołując menu *Projekt/Save Config*.

### Projekty przykładowe

Zanim zaczniemy pisać własne programy proponuję pobawić się przykładami. Polecam efektywny *Enkoder 8 na 3*, który znajduje się na płycie CD-EP1/2002B oraz na stronie internetowej EP w dziale *Download>Dokumentacje*.

W menu wybieramy *GAL-Asembler*, a następnie z podmenu *Assemble file* lub naciskamy klawisze *Amiga+A*. W oknie, które ukaże się, możemy zaznaczyć jakie pliki mają być generowane, i tak:

- *JEDEC* - koniecznie zaznaczyć, opcja generuje plik z rozszerzeniem *.jed*. Ten plik jest wymagany do zaprogramowania GAL-a.
- *Fuse-Matrix* - plik z rozszerzeniem *.fus*, pokazujący miejsca „przepaleń“ matrycę bezpieczników, a właściwie matrycę łączników konfigurujących.
- *Chip-Diagram* - plik z rozszerzeniem *.chp*, zawierający widok obudowy układu z etykietami pinów, czyli z nazwami sygnałów przyporządkowanymi wyprowadzeniom układu GAL.
- *Pin-Diagram* - plik z rozszerzeniem *.pin*, tabelka z etykietami i numerami pinów, do których są przypisane.
- *Autosave* - polecam zaznaczyć, dzięki temu nie będziemy każdorazowo pytani o to, czy zapisać wygenerowany plik.
- *adjust type of gal* - polecam zaznaczyć, dzięki temu program ustawi w menu taki typ GAL-a, jaki jest ustawiony w pliku.

Po naciśnięciu OK pojawi się okno wyboru plików. Wskazujemy plik *Enkoder 8 na 3.pld* z katalogu *Examples*. Jeśli kompilacja przebiegła pomyślnie, w oknie pojawi się komunikat *assembled file*, a w pustych dotychczas okienkach pojawią się nazwy etykiet (rys. 8). Wybieramy teraz z menu *GAL/Program* lub naciskamy klawisze *Amiga+P*. Pojawi się pytanie o typ układu GAL. Jeśli zaznaczyliśmy wcześniej opcję *adjust type of gal*, automatycznie zostanie wybrany GAL20V8. Jeśli nie zaznaczyliśmy, to wybieramy ten właśnie typ układu. Następnie umieszczamy układ w podstawce ZIF i naciskamy przycisk *Continue*. W oknie, które się pojawi, wybieramy wygenerowany wcześniej plik *Enkoder 8 na 3.jed*. Jeśli pamięć konfiguracji układu GAL będzie czysta, układ zostanie zaprogramowany, jeśli nie - pojawi się okno z komunikatem, w którym możemy wybrać czy chcemy wymazać starą zawartość, czy rezygnujemy z programowania. Po pomyślnym programowaniu automatycznie jest przeprowadzana weryfikacja.

Działanie układu można przetestować za pomocą programatora. Wybieramy z menu *Tools/GAL-Checker*. Naciskając przycisk przy oknach z nazwami etykiet możemy ustawić poziom H lub L na wejściu GAL. Niektóre wyprowadzenia z prawej strony układu mogą być wejściami lub wyjściami. O tym jaką funkcję wyprowadzenia wybieramy decyduje przycisk z literą „O“ - *output* lub „I“ - *input*. Wyprowadzenia, które są wyjściami, mogą przyjąć poziom H, L lub stan wysokiej impedancji oznaczony literką „Z“. Działanie enkodera jest podobne jak układu 74148. Pojawienie się poziomu logicznego L na wejściu IO powoduje ustawienie na wyjściach A, B, C poziomu L (liczba 0). Poziom L na I1 ustawi na wyjściu liczbę 1, itd. Gdy dwa wejścia znajdują się w stanie aktywnym, ustawiona zostanie liczba tego, które ma mniejszą wartość np.: przy ustawieniu wejść I6 i I3, na wyjściach ustawi się liczba 3. Wyprowadzenie En przyjmuje stan logiczny, gdy którekolwiek z wejść znajdzie się w stanie niskim. Dzięki temu można określić przypadek, w któ-

### WYKAZ ELEMENTÓW

#### Rezystory

- R1, R2: 1kΩ
- R3...R13, R16, R19...R26: 10kΩ
- R14: 1Ω
- R15: 27kΩ
- R17: 4,7kΩ
- R18: 4,7Ω
- R27: 22Ω
- R28: drabinka 8\*2,2kΩ
- R29...R32: nie istnieje (usunięto podczas modyfikacji projektu)
- R33: 22kΩ
- R34: 220Ω
- R35...R39: 1kΩ
- R40...R44: helitrym 2,0kΩ
- REZ1...REZ6: do przyszłych zastosowań

#### Kondensatory

- C1: 100pF
- C2: 4,7μF
- C3, C5...C14, C16: 100nF
- C4: 220μF/25V
- C15: 10μF
- C17: 100μF/16V

#### Półprzewodniki

- D1...D4: 1N4148
- LED: dowolna dioda LED
- T1, T3: BC557
- T2, T4, T5: BC547
- IC1, IC3...IC5: 4094
- IC2: 4555
- IC6: nie istnieje (usunięto podczas modyfikacji projektu)
- IC7: 4021
- IC8: 74LS06
- IC9: TL497
- IC10: 74LS145
- IC11: 74HC126
- IC12: 7805

#### Różne

- K1, K2: EDR201C1200 (przekaznik kontaktronowy w obudowie DIL 12V)
- L1: dławik 100μH
- CON1: ARK-2
- TEXT00L: Podstawka zatraskowa 24 pin 0,3"
- X1: Gniazdo IDC34
- Wtyk IDC34 na kabel flat
- Taśma flat 34
- DB25-M na kabel FLAT

rym stan aktywny pojawił się na wejściu I7.

Enkoder taki można zastosować jako układ wejściowy do ośmioprzyciskowej klawiatury. Na wyjściach A, B, C pojawia się informacja o numerze naciśniętego klawisza. Wyjście EN może

generować sygnał strobujący. Układ ten może też pełnić funkcję dekodera priorytetu przerwania. Linia EN połączona jest wtedy z linią INT CPU. Wystąpienie jednego z ośmiu przerwania wywoła obsługę przerwania przez CPU, w której na podstawie stanu wyjść A, B, C dekodera nastąpi skok do odpowiedniej procedury. Najwyższy priorytet

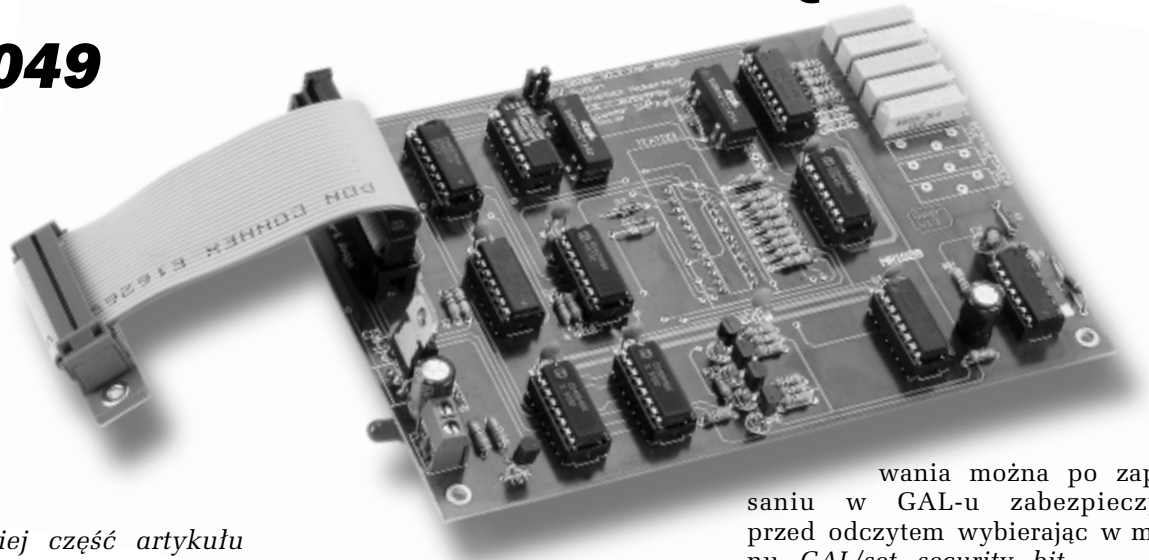
będzie miało przerwanie o numerze 0. Wejście przerwania CPU musi reagować na poziom. Spowodowane jest to tym, że gdy pojawią się dwa przerwania równocześnie, po zakończeniu obsługi tego o wyższym priorytecie, wyjście En cały czas będzie na poziomie L. Poziom ten zmieni się dopiero po skończeniu obsługi drugiego przerwania.

Z opcji testowania wychodzimy wybierając w menu *GAL-Checker/Exit* lub naciskając *Amiga+X*.  
**Sławomir Skrzyński, AVT**  
**slawomir.skrzynski@ep.com.pl**

*Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/styczen02.htm> oraz na płycie CD-EP01/2002B w katalogu PCB.*

# GALER - programator GAL-i do Amigi, część 2

## AVT-5049



W drugiej części artykułu przedstawiamy język opisu układów implementowanych w strukturach PLD.

Omawiamy również oprogramowanie sterujące pracą programatora oraz jego obsługę.

### Programy dla GALERA

Zabierzmy się teraz do opisanego układu, który chcemy zrealizować w GAL-u. Uruchamiamy edytor z menu (*Tools/Call Editor*) lub naciskając prawą *Amigę+E*. Następnie wpisujemy opis przykładowego układu (**list. 1**). Jak widać na listingu, w pierwszej linii określamy typ układu GAL, w drugiej sygnaturę (maks. 8 znaków), która zostanie wpisana do układu. Sygnaturę można odczytać wybierając z menu *GAL-Disassembler/Read signature*.

W kolejnych liniach wpisujemy etykiety pinów (identyfikatory sygnałów wejściowych i wyjściowych) poczynając od 1. Niżej znajdują się równania opisujące działanie układu. Wszystko co znajdzie się za słowem kluczowym *Description* nie będzie interpretowane. W każdej linii można umieścić komentarz po znaku średnika. Zestaw dostępnych komend kompilatora jest dość skromny. Mamy do dyspozycji następujące operacje:

- OR - w opisie źródłowym oznaczana jako „#” lub „+“,
- AND - w opisie źródłowym oznaczana jako „&” lub „\*“,
- NOT - w opisie źródłowym oznaczana jako „!” lub „/“.

Za pomocą powyższych operacji można zapisać dowolną funkcję przełączającą. Otrzymamy po skompilowaniu plik do programo-

wania można po zapisaniu w GAL-u zabezpieczyć przed odczytem wybierając w menu *GAL/set security bit*.

Wyjścia układu GAL mogą być trójstanowe. Wówczas nazwę wyjście deklarujemy z rozszerzeniem *.T*. Potrzebna jest też deklaracja wejścia sterującego wyjściem trójstanowym. Przy nazwie sygnału wpisuje się wówczas rozszerzenie *.E*. Przykład takiego zapisu można zobaczyć na **list. 1**. Wyjście ANDT przyjmie stan trzeci, jeśli wejście *En* znajdzie się na poziomie L.

Wyjścia GAL-a mogą być typu rejestrowego. Pozwala to tworzyć liczniki, zatraski, rejestry przesuwne itp. Przykład opisu licznika zamieszczono na **list. 2**.

Synchronicznie z narastającym zboczem sygnału *Clock* zwiększa się zawartość licznika o 1. Przy poziomie H na wejściu *Clear* i narastającym zboczem zegarowego sygnału zeruje się licznik, natomiast przy wysokim poziomie na wejściu *Set* i aktywnym zboczem *Clock* zostaną przepisane dane z wejść usta-



Rys. 11. Okno z widokiem rozmieszczenia wyprowadzeń projektowanego układu.

```
List. 1.
GAL20V8           ;Typ układu
Bramki           ;Sygnatura (max. 8-char.)
NC A B C D E F G H I J GND ;Deklaracja pinów
NC EN ANDT NC EXNOR EXOR NOR OR NAND AND NC VCC

AND = A*B           ;Równanie opisujące bramkę AND
/NAND = C*D         ;Równanie opisujące bramkę NAND
OR = E + F          ;Równanie opisujące bramkę OR
/NOR = G + H        ;Równanie opisujące bramkę NOR
EXOR = I*J + /I*/J  ;Równanie opisujące bramkę EXOR
/EXNOR = I*J + /I*/J ;Równanie opisujące bramkę EXNOR

ANDT.T = A*B        ;Bramka and w wyjście trójstanowym
ANDT.E = EN         ;Sterowanie wyjściem trójstanowym

DESCRIPTION:
Na podstawie artykułu "Układy PLD" (EP 19/93 str.28) List 2a
*: AND
+: OR
/: NEGACJA
```

wiających na wyjścia. Wejście *Clear* ma wyższy priorytet niż wejście *Set*. Wyjścia rejestrowe mogą przyjąć stan trzeci (wysokiej impedancji) za sprawą sygnału */OE EN*. Na list. 3 pokazano opis układu 74574. Zaletą GAL-a jest m.in. to, że można dość dowolnie rozmieszczać wyprowadzenia projektowanego układu, co uczyniono w tym przykładzie. Dzięki możliwości prawie dowolnego rozmieszczenia wyprowadzeń można znacznie uprościć układ ścieżek obwodu drukowanego.

W pliku zawierającym archiwum (dostępny w Internecie na stronie [www.ep.com.pl](http://www.ep.com.pl) w dziale *Download* oraz na CD-EP1/2002B) w katalogu „Examples“ znajdują się wszystkie przedstawione tu opisy i wiele innych. Znajdziecie tam m.in. opis transkodera z kodu BCD na kod wyświetlacza 7-segmentowego. Dzięki możliwości zmiany rozmieszczenia wyprowadzeń, zarówno wejściowych jak i wyjściowych, bez problemu można do transkodera podłączyć dowolny licznik i dowolny wyświetlacz, zachowując bezpośrednie połączenia na płytce. Ponadto można dowolnie zmieniać wygląd znaków.

W trybie rejestrowym umieszczenie wyprowadzeń dla sygnałów *Clock* i *Enable* jest ściśle określone. W tab. 1 zestawiono wszystkie możliwe funkcje pełnione przez poszczególne wyprowadzenia w różnych trybach.



Rys. 12. Widok okna optymalizatora równań.



Rys. 13. Sposób umieszczenia układu w podstawce.

mi mogą przechodzić w stan trzeci za sprawą sygnału z dowolnego wejścia i dowolnym poziomem (aktywny poziom niski lub wysoki).

Zostało do opisanie jeszcze kilka funkcji menu, o których nie wspominałem. Menu „project/About“ wyświetli okno z adresem kontaktowym do autora programu. W menu „Gal“ można wybrać typ obsługiwanego układu. Menu „GAL/Copy“ kopiuje układ. Najpierw jesteśmy proszeni o umieszczenie układu źródłowego, a po odczycie jego zawartości - docelowego. „GAL/Erase“ kasuje zawartość pamięci układu, „GAL/Compare“ umożliwia porównanie zawartości dwóch układów. „GAL/Blank test“ sprawdza czy układ nie jest zapisany, „GAL/Test security bit“ sprawdza stan bitów zabezpieczających. W „GAL/Write Access“ możemy ustawić jakie operacje będą wykonywane:

- przy programowaniu układu:
  - „with blank test“ - sprawdzanie przed programowaniem czy czysty - polecam ustawić tę opcję, ponieważ przed programowaniem GAL-a musi być on kasowany;
  - „with verify“ - weryfikacja poprawności zapisu;
- podobnie przy kopiowaniu,
- przed kasowaniem: „with blank test“ - sprawdzenie, czy układ jest czysty.

Standardowo (Mode 1) wyprowadzenia mogą też pełnić funkcję wejść, poza dwoma wyprowadzeniami, które mogą być tylko wyjściami. Jeśli wykorzystamy możliwość pracy trójstanowej (Mode 2), wszystkie wyprowadzenia mogą być wejściami poza dwoma. W trybie rejestrowym (Mode 3) wejście sygnału zegarowego i sterowania wyjściem trójstanowym jest przypisane na stałe do wyprowadzeń 1 i 13 dla GAL20V8 (11 dla GAL16V8). Przerzutniki są przełączone zboczem narastającym. Jeśli chcielibyśmy, aby były przełączane zboczem opadającym, należałoby wprowadzić w układzie negator i jego wyjście połączyć z wyprowadzeniem 1 układu. Każde wyjście może być wyjściem rejestrowym trójstanowym, wyjściem kombinacyjnym trójstanowym lub wyjściem kombinacyjnym. Należy zaznaczyć, że dla wyjść rejestrowych stan trzeci jest wyłączany sygnałem z wyprowadzenia 13 dla GAL20V8 (11 dla GAL16V8) - wyjścia w stanie trzecim przy poziomie H. Wyjścia, które nie są wyjściami rejestrowymi

```
List. 2.
GAL20V8           ; 4-Bit-Counter      first line: used GAL
Counter          ;                      second line: any text (max. 8 char.)
Clock D0         D1 D2 D3 Set Clear NC NC NC NC GND
/OE NC           NC NC NC Q3 Q2 Q1 Q0 NC NC NC VCC

Q0.R = /Clear * Set * D0
      + /Clear * /Set * /Q0

Q1.R = /Clear * Set * D1
      + /Clear * /Set * /Q1 * Q0
      + /Clear * /Set * Q1 * /Q0

Q2.R = /Clear * Set * D2
      + /Clear * /Set * Q2 * /Q1
      + /Clear * /Set * Q2 * /Q0
      + /Clear * /Set * /Q2 * Q1 * Q0

Q3.R = /Clear * Set * D3
      + /Clear * /Set * Q3 * /Q2
      + /Clear * /Set * Q3 * /Q1
      + /Clear * /Set * Q3 * /Q0
      + /Clear * /Set * /Q3 * Q2 * Q1 * Q0

DESCRIPTION
this is a 4-Bit-Counter
registered outputs are signed with the postfix.R
Pin 'Set' HIGH while clock signal (LOW-HIGH) at pin 'Clock':
    load Q0-Q3 with state of D0-D3
Pin 'Clear' HIGH while clock signal:
    Q0-Q3 are cleared

Clock signal while pins 'Set' and 'Clear' are LOW:
increment counter
```



Tab. 1.

GAL16V8:								
Mode 1	Mode 2	Mode 3	Mode 1	Mode 2	Mode 3	Mode 1	Mode 2	Mode 3
In	In	Clock	1	20	+5V	+5V	+5V	
In	In	In	2	19	In/C	T*	In/T/R	
In	In	In	3	18	In/C	In/T	In/T/R	
In	In	In	4	17	In/C	In/T	In/T/R	
In	In	In	5	16	C	In/T	In/T/R	
In	In	In	6	15	C	In/T	In/T/R	
In	In	In	7	14	In/C	In/T	In/T/R	
In	In	In	8	13	In/C	In/T	In/T/R	
In	In	In	9	12	In/C	T*	In/T/R	
GND	GND	GND	10	11	In	In	/OE	
GAL20V8:								
Mode 1	Mode 2	Mode 3	Mode 1	Mode 2	Mode 3	Mode 1	Mode 2	Mode 3
In	In	Clock	1	24	+5V	+5V	+5V	
In	In	In	2	23	In	In	In	
In	In	In	3	22	In/C	T*	In/T/R	
In	In	In	4	21	In/C	In/T	In/T/R	
In	In	In	5	20	In/C	In/T	In/T/R	
In	In	In	6	19	C	In/T	In/T/R	
In	In	In	7	18	C	In/T	In/T/R	
In	In	In	8	17	In/C	In/T	In/T/R	
In	In	In	9	16	In/C	In/T	In/T/R	
In	In	In	10	15	In/C	T*	In/T/R	
In	In	In	11	14	In	In	In	
GND	GND	GND	12	13	In	In	/OE	

Legenda:

- In: wejście
- C: wyjście
- T: wyjście trójstanowe
- T\*: wyjście trójstanowe (wyprowadzenie to nie może być wejściem)
- R: wyjście rejestrowe
- Clock: wejście zegarowe dla komórek rejestrowych
- /OE: wejście sterujące pracą buforów trójstanowych dla wyjść rejestrowych

Menu „GAL-Disassembler/Gal-In-fo“ wyświetla informację o producencie układu i zastosowanym algorytmie programowania. „GAL-Disassembler/Generate JEDEC file“ odczytuje zawartość GAL-a i zapisuje do pliku z rozszerzeniem „jed“. Plik ten może posłużyć do zaprogramowania innego układu lub do desasemblacji. „GAL-Disassembler/JEDEC file parametr“ pozwala ustawić opcje zapisu pliku (czy ma dodać sumy kontrolne, itp). Program jest wyposażony w deassembler. Z plików z rozszerzeniem „jed“ (plik wynikowy do programowania GAL-a) generuje pliki źródłowe z rozszerzeniem „pld“. Po wybraniu tej pozycji menu, pojawi się okno z możliwością przypisania etykiet (identyfikatorów sygnałów) do fizycznych wyprowadzeń układu (patrz rys. 11). W menu „Tools“ dostępne jest polecenie „Clear pinenames“, które kasuje etykiety z okna programu. Natomiast polecenie „Show pinenames“ wyświetla

etykiety. Interesującą opcją jest optimizer wybierany z menu „Tools/Optimizer“. Jeśli projektowany układ nie mieści się w GAL-u, ten program może nas uratować. Po wskazaniu pliku do optymalizacji wyświetli się okno (rys. 12), w którym jest wyświetlone równanie w postaci w jakiej znajduje się w pliku źródłowym (*old equation*) oraz w postaci zoptymalizowanej (*optimized equation*), poniżej zostanie wyświetlona zużyta na ten cel liczba bramek matrycy AND i OR. Jeśli po optymalizacji liczba bramek jest mniejsza, zmianę wartość zaakceptować naciskając „use it“. Po optymalizacji wybieramy plik do zapisu.

**Uwagi końcowe**

Program posiada wbudowaną pomoc. Jeśli nie wiemy co dany requester, gadżet, czy opcja menu robi, wystarczy wskazać opcję w menu (czy też okno/gadżet) myszką i nacisnąć klawisz „Help“

(standardowe wywołanie pomocy w Amiga OS). Wtedy uruchomi się program AmigaGuide z wybranym rozdziałem, w którym opisano interesujące nas zagadnienie.

Ponieważ podstawka programująca posiada 24 styki, a układy GAL16V8 zaledwie 20 wyprowadzeń, należy zwrócić uwagę na montaż tych układów w podstawie. Sposób umieszczenia takiego układu w podstawie przedstawiono na rys. 13.

Do programu sterującego pracą programatora dołączono pliki z kodami źródłowymi. Program można „ściągnąć“ z Aminetu. Jest on shareware'owy. Rejestrację należy przeprowadzić u autora:

Christian Habermann  
Asamstr. 17  
85356 Freising, Germany  
Christian.Habermann@t-online.de

Za pomocą programatora prezentowanego w artykule zaprogramowałem już wiele GAL-i. Przy jakichkolwiek problemach proszę o kontakt.

**Sławomir Skrzyński, AVT**  
slawomir.skrzynski@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/luty02.htm> oraz na płycie CD-EP02/2002B w katalogu PCB.

List. 3.							
GAL20V8 ; 8-Bit-Latch							
74574 ;							
Clock	Reset	D0	D1	D2	D3	D4	D5
D6	D7	NC	GND	Q1	Q2	Q3	Q4
/OE	NC	Q0	Q1	Q2	Q3	Q4	Q5
Q6	Q7	NC	VCC				
/Q0.R =	/D0						
/Q1.R =	+ /Reset						
/Q2.R =	+ /Reset						
/Q3.R =	+ /Reset						
/Q4.R =	+ /Reset						
/Q5.R =	+ /Reset						
/Q6.R =	+ /Reset						
/Q7.R =	+ /Reset						
DESCRIPTION							
Clock	1	24	VCC				
Reset	2	23	NC				
D0	3	22	Q7				
D1	4	21	Q6				
D2	5	20	Q5				
D3	6	19	Q4				
D4	7	18	Q3				
D5	8	17	Q2				
D6	9	16	Q1				
D7	10	15	Q0				
NC	11	14	NC				
GND	12	13	/OE				