Mikrokontroler 68HC908QT4A Mikrokontroler 68HC908QT4A klawiatura i linijka diodowa

Opis układów wykorzystanych w aplikacji

Układ 74LS164 jest rejestrem przesuwnym służącym do zamiany informacji szeregowej na równoległą. Układ, którego symbol logiczny pokazuje rysunek 1, posiada dwa wejścia danych A i B (w prostych aplikacjach zwarte), wejście zegarowe CP, wejście zerujące MR oraz wyjścia równoległe Q0-Q7. Pojawienie się narastającego zbocza na wejściu CP powoduje wpisanie informacji z wejść danych AB na wyjście Q0, równocześnie dotychczasowy stan wyjścia Q0 jest przepisywany na wyjście Q1, z wyjścia Q1 na Q2 itd. Ośmiokrotne pojawienie się narastającego zbocza zegarowego powoduje wpisanie kolejnych 8-miu stanów z wejść AB na wyjścia Q0-Q7.

LOGIC SYMBOL



Rys.1. Symbol logiczny układu 74LS164 [http://dpjyy.myvtc.edu.cn/]

Układ 74LS165 jest rejestrem przesuwnym równoległo-szeregowym, służącym do zamiany informacji wprowadzonej równolegle do rejestru na szeregową. Symbol logiczny układu pokazano na rysunku 2. Układ posiada wejście ładowania informacji równoległej PL. Podanie stanu niskiego na to wejście powoduje zatrzaśnięcie informacji równoległej podanej na wejścia P0-P7 w przerzutnikach rejestru przesuwnego. Pojawienie się narastającego zbocza na wejściu zegarowym CP (podwójne z możliwością bramkowania, w normalnej pracy oba piny 2 i 15 zwarte) powoduje przesuwanie informacji w rejestrze przesuwnym o jeden od wyjścia P0 do P7. Przesunięta informacja pojawia się na wyjściu Q7 (zanegowana na /Q7), równocześnie na wyjście P0 wpisywana jest informacja z wejścia Ds.

LOGIC SYMBOL



V_{CC} = PIN 16 GND = PIN 8

Rys.2. Symbol logiczny układu 74LS164 [http://susta.cz/fel/74/pdf/sn_74165.pdf]

Aplikacja użyta w ćwiczeniu

Oba rejestry przesuwne zostały wykorzystane, w wykorzystywanej w ćwiczeniu aplikacji mikrokontrolera 68HC908QT4A, i służą do zwiększenia ilości linii portów wyjścia (do sterowania 16-toma diodami LED) oraz linii wejścia (8-miu przycisków monostabilnych).

Schemat podłączenia do mikrokontrolera zewnętrznych rejestrów przesuwnych służących do obsługi 8 wejść oraz 16-tu wyjść pokazuje rysunek 3.



Rys.3. Schemat podłączeń do mikrokontrolera 68HC908QT4A

Jak widać z rysunku do sterowania diod o kolorze czerwonym i zielonym przewidziano dwa rejestry przesuwne 74LS164. Wpisywanie do rejestru jedynki logicznej "1" poprzez linię PTA3 mikrokontrolera powoduje zapalenie diody w kolorze zielonym, poprzez linię PTA4 diody czerwonej. Wpisanie jedynki logicznej "1" do obu rejestrów (linie PTA3, PTA4) powoduje iż dioda świeci w kolorze pomarańczowym. Wpisanie stanu logicznego sterującego linią wyjściową Q7 następuje jako pierwsze, stan logiczny sterujący linią Q0 jest wpisywany jako ostatni.

Podobnie przy odczycie klawiatury stan klawisza na wejściu D7 jest odczytywany jako pierwszy a stan wejścia D0 jako ostatni.

Program główny, realizujący reakcję na naciśnięcie klawiatury

Na rysunku 4 pokazano schemat blokowy pętli głównej programu, którego funkcją jest przepisywanie stanu wyjść przycisków umieszczonych na płycie czołowej modułu we-wy na rejestr sterujący diodami elektroluminescencyjnymi, z zastosowaniem konwersji sygnałów cyfrowych w zależności od zadania zaproponowanego w ćwiczeniu. Na rysunku pokazano również składnię assemblera realizującego pokazany algorytm programu.

Kolejne etapy głównej pętli programowej to:

- przygotowanie mikrokontrolera do pracy, programowanie układów peryferyjnych- INIT,
- odczyt stanu klawiatury-IN,
- konwersja odczytu- CONV,
- wyświetlenie wyniku- OUT,
- opóźnienie programowe TIME,
- skok do początku pętli poza czynności inicjalizujące.

Program główny



Rys.4. Schemat blokowy pętli głównej i sposób jego realizacji.

Procedury INIT, IN I CONV

Procedura INIT

Podstawowymi czynnościami po włączeniu mikrokontrolera są:

- wyłączenie układów nadzoru wykonywania programu na czas uruchamiania programów i testów,
- ustawienia odpowiednich stanów początkowych na wyjściach,
- ustawienie odpowiednich linii jako wyjścia lub jako wejścia,

programowanie układów czasowo-licznikowych, systemu przerwań, przetworników, interfejsów, itp.
W związku z powyższym podprogram INIT:

- Wyłącza układ nadzoru wykonywania programu WATCH-DOG, poprzez zapisanie odpowiedniej liczby do rejestru CONFIG 1.
- Ustawia linie portu PTA mikrokontrolera 68HC908 w odpowiednie stany początkowe (nieaktywne dla linii zegarowych i wejścia PL układu 74LS165,). Następuje to poprzez zapisy "0" i "1" logicznych na odpowiednie linie portu.
- Ustawia kierunek linii portu poprzez zapis "0" i "1" logicznych w rejestrze kierunku DDRA, które z tych linii są liniami wejścia, które liniami wyjścia.

Algorytm podprogramu INIT pokazano na rysunku 5.

Procedura IN

Rysunek 5 pokazuje również sposób realizacji procedury odczytu stanu naciśnięcia klawiatury IN. Początkową czynnością jest załadowanie informacji o stanie naciśnięcia klawiszy ("0"- wciśnięty) do wejść równoległych poprzez wygenerowanie impulsu zerowego na linii PL (PTA0). Następnie w zależności od stanu wejścia PTA2 flaga Carry jest zerowana lub ustawiana i jej stan jest wsuwany do komórki o adresie \$90. Po tym następuje wygenerowanie pojedynczego impulsu zegarowego na linii PTA1. Czynność ta jest powtarzana ośmiokrotnie w pętli programowej odliczanej zwartością akumulatora, z wykorzystaniem skoku DBNZA (zmniejsz akumulator o jeden i skocz jeśli nie 0). Wynik odczytu klawiatury jest zapisywany w komórce pamięci o adresie \$90 (w kodzie HEX).

Procedura CONV

Po odczycie stanu klawiatury zawartość komórki pamięci będzie przepisywana bez, lub po modyfikacji do komórki pamięci o adresie \$91 w procedurze CONV. Procedura CONV będzie modyfikowana zgodnie z zadaniami wyznaczonymi na końcu niniejszej instrukcji.



Rys.5. Procedura inicjalizacji mikrokontrolera INIT, odczytu stanu klawiatury IN, konwersji wyniku odczytu przed zapisaniem wyniku na diody świecące CONV.

Procedury OUT i TIME

Kolejne procedury, których algorytm oraz sposób realizacji pokazuje rysunek 6 służą do wyświetlenia wyniku konwersji (OUT) oraz opóźnienia czasowego przed kolejnym odczytem klawiatury (TIME).

Procedura CONV

Procedura OUT służy do wyświetlenia wyniku konwersji zapisanego w komórce o adresie \$91 (kod HEX). Wynik jest wyświetlany przy pomocy diod o kolorze czerwonym (używany jest bit PTA4, bit PTA3=0- diody zielone wygaszone). Pokazana procedura działa na diodach czerwonych, wykorzystując linię PTA4. Zawartość kolejnych bitów komórki \$91 jest wysuwana na bit Carry. W zależności od stanu tego bitu linia PTA4 jest ustawiana lub kasowana ("1"- dioda świeci). Po ustaleniu stanu logicznego na linii PTA4, generowany jest pojedynczy impuls zegarowy na linii PTA1. Czynność ta jest powtarzana w pętli programowej odliczanej zawartością rejestru akumulatora.

Procedura TIME

Procedura opóźniająca nazwana TIME wprowadza do programu opóźnienie czasowe wynikające z wielokrotnego wykonania instrukcji zmniejszenia zawartości komórki pamięci i skoku jeśli zawartość tej komórki nie wynosi zero. Czas wykonania każdej instrukcji w mikrokontrolerze jest ściśle określony i jego wielokrotne wykonanie powoduje wprowadzanie tzw. opóźnień programowych. Do tworzenia takich opóźnień służy rozkaz o składni DBNZ- adres dowolnej komórki pamięci, składnia ta oznacza zmniejsz o 1 i skocz jeśli nie zero. Pomnożenie tego czasu następuje poprzez umieszczenie pętli w pętli.



Tek

Tek



Zmniejszenie zawartości lokacji \$A0 o 1 i sprawdzenie czy zawartość \$A0 = 0

Powrót do programu głównego

Nie

Rys.6. Procedura wyświetlania OUT, opóźnienie programowe TIME

Zmniejszenie zawartości lokacji \$A1 o 1 i sprawdzenie czy zawartość \$A1 = 0

Zadania do wykonania.

Zadanie 1:

Dokonać zmian w programie tak, aby po wciśnięciu przycisku następowało świecenie przeciwległej diody LED (w kolorze czerwonym).

Zadanie 2:

Dokonać zmian w programie tak, aby po wciśnięciu przycisku następowało świecenie przeciwległej diody LED (w kolorze zielonym).

Zadanie 3:

Dokonać zmian w programie tak, aby po wciśnięciu przycisku następowało świecenie przeciwległej diody LED (w kolorze pomarańczowym).

Zadanie 4: (ułożyć obudowę z elementami wykonawczymi tak, aby diody LED znajdowały się "na górze", a przyciski "na dole")

Dokonać zmian w programie tak, aby po wciśnięciu kilku przycisków następowało wyświetlenie ich ilości w kodzie binarnym, na diodach LED (w kolorze czerwonym), z wyrównaniem do prawej, kolejne wagi to 2^3 , 2^2 , 2^1 , 2^0 . (2^0 skrajny bit).

Zadanie 5: (ułożyć obudowę z elementami wykonawczymi tak, aby diody LED znajdowały się "na górze", a przyciski "na dole")

Dokonać zmian w programie tak, aby po wciśnięciu kilku przycisków równocześnie, świeciła dioda (w kolorze czerwonym) odpowiadająca przyciskowi położonemu najdalej na lewo.

Zadanie 6: (ułożyć obudowę z elementami wykonawczymi tak, aby diody LED znajdowały się "na górze", a przyciski "na dole")

Dokonać zmian w programie tak, aby po wciśnięciu kilku przycisków równocześnie, trzy diody od strony prawej - wagi 2^3 , 2^2 , 2^1 , 2^0 (2^0 skrajne), świecąc w kolorze czerwonym podawały numer przycisku położonego najdalej na lewo. Skrajny lewy bit informuje o wciśnięciu jakiegokolwiek przycisku, jak pokazano na rysunku poniżej. Gdy nic nie wciśnięte, wszystkie LED-y wygaszone.

