

Wojciech MIELCZAREK

Instytut Informatyki Czasu Rzeczywistego
Politechnika Śląska

PROGRAMOWANIE PAMIĘCI STAŁYCH REPROGRAMOWALNYCH

Streszczenie. Podstawowym elementem współcześnie produkowanych mikroprogramowanych urządzeń cyfrowych jest elektrycznie programowana i reprogramowana pamięć stała EPROM, która najczęściej pełni rolę pamięci programów określających cykl pracy tych urządzeń. Wobec powszechnego zastosowania pamięci EPROM bardzo ważne staje się zagadnienie ich prostego, szybkiego i niezawodnego programowania. Niniejszy artykuł omawia zasady programowania pamięci stałych MOS EPROM typu FAMOS oraz przedstawia konstrukcję programatora przeznaczonego do programowania pamięci INTEL 1702A.

1. Wstęp

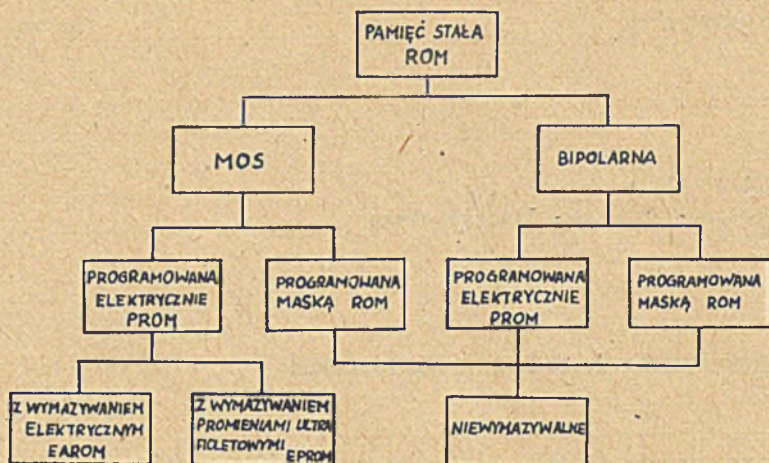
Pamięć stała ROM (Read Only Memory) jest obecnie powszechnie stosowanym układem cyfrowym, którego zasadniczą cechą jest to, że zapisaną w nim informację można jedynie odczytywać. Pamięci ROM posiadają poza tym ważną zdolność przechowywania informacji mimo wyłączenia napięcia zasilania i jako takie znalazły szerokie zastosowanie:

- w komputerach do realizacji mikroprogramów oraz w innych mikroprogramowanych urządzeniach cyfrowych,
- w kalkulatorach do generowania funkcji standardowych,
- w monitorach i wyświetlaczach jako translatory kodów,
- jako generatory funkcji przełączających (np. dekodery, kodery, układy sprawdzania parzystości itp.).

Zasadniczo pamięci ROM występują w trzech rodzajach: programowane w czasie produkcji, programowane jednorazowo (bez możliwości zmian) przez użytkownika oraz reprogramowalne, tzn. takie, których zawartość może być zmieniana wielokrotnie przez użytkownika.

Na rys. 1 przedstawiono ogólny podział półprzewodnikowych pamięci stałych dokonany ze względu na sposoby ich programowania i wymazywania.

Szczególą popularność zyskały pamięci programowane samodzielnie przez użytkowników, tzw. PROM-y, wśród których wyróżnić należy PROM-y bipolarne, MOS EPROM-y i n MOS EAROM-y.



Rys. 1. Ogólny podział półprzewodnikowych pamięci stałych

PROM-y bipolarne są najczęściej wykonywane w technologii Schottky TTL i stąd charakteryzują się przede wszystkim dużą szybkością działania (czas dostępu od 30-90 ns) przy stosunkowo dużym prądzie pobieranym ze źródła napięcia zasilania (65-180 mA). Pamięci te programowane są jednorazowo przez użytkownika poprzez przepalenie ścieżek metalizacji za pomocą impulsów prądu elektrycznego. PROM-y bipolarne szczególnie nadają się do współpracy z szybkimi mikroprocesorami.

Pamięci MOS EPROM są z kolei pamięciami reprogramowalnymi. Powszechnie stosowane są EPROM-y wykonane w technologii FAMOS, tanie, łatwo programowalne, stosunkowo szybkie (jakkolwiek 5-10 razy wolniejsze od PROM-ów bipolarnych) i pobierające małą moc ze źródła napięcia zasilania. Ich podstawową wadą jest konieczność stosowania kilku napięć zasilających (z wyjątkiem pamięci INTEL 2716 wymagającej tylko napięcia zasilającego 5 V). Pamięci MOS EPROM stały się szczególnie popularne i szeroko stosowane przy konstrukcji urządzeń prototypowych, gdzie często ulegają zmianie żądania dotyczące zawartości pamięci. Programowanie pamięci EPROM typu FAMOS odbywa się na drodze elektrycznej, a kasowanie zawartości pamięci odbywa się przez jej naświetlenie promieniowaniem ultrafioletowym lub promieniami X.

Pamięci n MOS EAROM są pamięciami, których programowanie jak i wymazywanie odbywa się na drodze elektrycznej. Umożliwia to selektywne programowanie poszczególnych bitów pamięci bez konieczności uprzedniego kasowania zawartości całej pamięci. Pamięci n MOS EAROM są jednak zbyt wolne dla większości zastosowań w urządzeniach czasu rzeczywistego (czas dostępu 0,95-5 μ s) i obecnie jeszcze stosunkowo drogie.

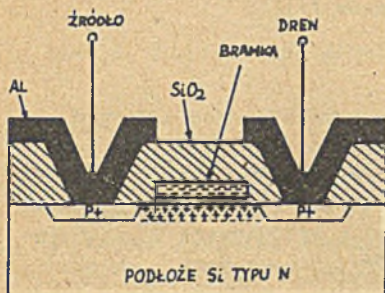
Niniejszy artykuł omawia zasady programowania pamięci stałych MOS PROM typu FAMOS oraz przedstawia prosty programator przeznaczony do programowania pamięci INTEL 1702A. Zagadnienia związane z programowaniem pamięci

typu FAMOS wydają się obecnie szczególnie ważne ze względu na ich stale rosnącą popularność (70% produkcji programowanych pamięci stałych) oraz przewidywaną produkcję pamięci 1702A w Polsce.

2. Zjawiska fizyczne zachodzące podczas programowania komórki pamięci stałej FROM FAMOS

Komórkę pamięci stałej FROM FAMOS stanowi p-kanalowy tranzystor MOS z bramką krzemową zawieszoną w tlenku krzemu, który jest dobrym materiałem dielektrycznym. Grubość tlenku krzemu pod bramką wynosi $0,1 \mu\text{m}$ a nad nią około $1 \mu\text{m}$. Bramka nie posiada żadnego kontaktu elektrycznego i stąd bywa również nazywana "pływającą" (floating gate). Programowanie komórki odbywa się poprzez podanie odpowiednio wysokiego (powyżej 30 V) ujemnego napięcia pomiędzy źródłem a drenem, które

powoduje zjawisko lawinowego przebicia pomiędzy drenem a podłożem. Część elektronów przepływających przez złącze osiąga energię wystarczającą aby pokonać cienką warstwę dielektryka oddzielającą bramkę od podłoża i przedostać się do bramki, co powoduje powstanie inwersyjnego ładunku w warstwie krzemu znajdującej się pod bramką. Utworzony tak przewodzący kanał między drenem a źródłem pozostaje nawet po usunięciu wysokiego "programującego" napięcia. Ładunek zgromadzony w bramce nie ma bowiem możliwo-



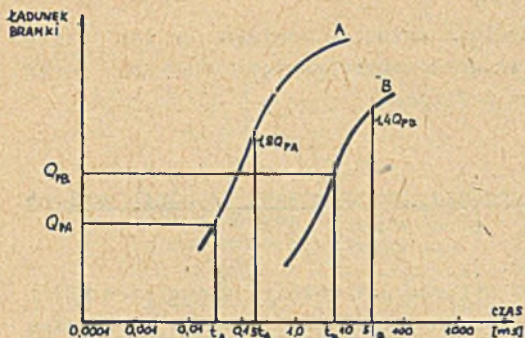
Rys. 2. Tranzystor FAMOS po zaprogramowaniu

ści odpływu ze względu na bardzo dobre właściwości dielektryczne otaczającej bramkę tlenku krzemu. Zaprogramowany tranzystor FAMOS będzie się więc charakteryzował stosunkowo małą rezystancją R_{SD} pomiędzy drenem a źródłem, w odróżnieniu od nie zaprogramowanego, w którym rezystancja R_{SD} jest bardzo duża.

Na rys. 2 przedstawiono tranzystor FAMOS po zaprogramowaniu.

3. Charakterystyki programowania komórek pamięci typu FAMOS

Charakterystyka programowania obrazuje proces gromadzenia ładunku w bramce tranzystora stanowiącego programowaną komórkę w zależności od czasu programowania, przy stałym napięciu programującym (najczęściej $U = -47 \text{ V}$). Szybkość gromadzenia ładunku w bramce, jak również maksymalna wartość ładunku, który może być w niej zakumulowany, zależą od napięcia przebicia lawinowego złącza p-n pomiędzy drenem a podłożem oraz od rezystancji R_{xy}



Rys. 3. Charakterystyki programowania dwóch przykładowych komórek pamięci 1702A

(rys. 4), jaką stanowi tranzystor układu adresowania danej komórki.

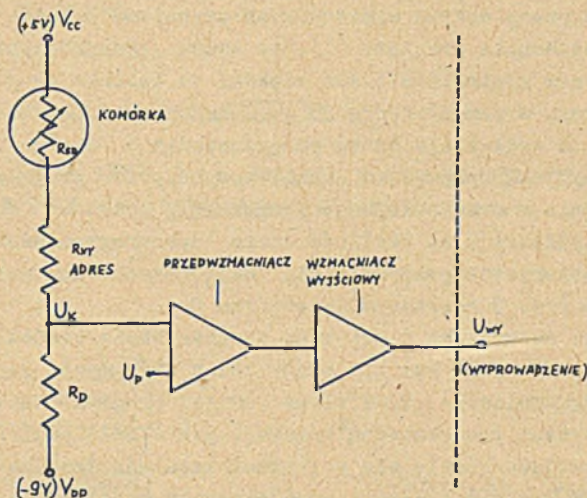
Na rys. 3 przedstawiono charakterystyki programowania dwóch komórek pamięci 1702A, z których A reprezentuje komórki charakteryzujące się dużym napięciem przebiccia lawinowego oraz małą rezystancją R_{xy} (kilkanaście Ω), a B komórki o stosunkowo małym napięciu przebiccia lawinowego i dużej rezystancji R_{xy} (rzędu kilkudziesięciu Ω). Komórki typu A stanowią około 98% komórek pamięci większości PROM-ów [1].

4. Zasady programowania pamięci PROM FAMOS

Na rys. 4 przedstawiono schemat zastępczy komórki pamięci w stanie odczytu. Komórka pamięci reprezentowana jest przez rezystancję R_{SD} , a jej układ adresowania przez R_{xy} . Wzmacnianie napięcia wyjściowego U_k z komórki realizowane jest przez przedwzmacniacz (który jednocześnie porównuje napięcie U_k ze stałym napięciem progowym U_p) oraz przez wzmacniacz wyjściowy. Jeżeli napięcie U_k jest mniejsze od U_p , napięcie U_{wy} jest ujemne i bit odczytywany jest jako nie zaprogramowany. Jeżeli natomiast U_k jest większe od U_p , napięcie wyjściowe staje się dodatnie i bit uznawany jest za zaprogramowany.

Stan zaprogramowania komórki należałoby zasadniczo określać poprzez pomiar rezystancji R_{SD} , co jest jednak ze względów technicznych niemożliwe. Użytkownik ma bowiem dostęp tylko do napięcia U_{wy} , a zatem może oceniać zaprogramowanie komórki jedynie poprzez obserwację U_{wy} .

Problemem pozostaje zawsze "bezpieczne" zaprogramowanie komórki, ponieważ może się zdarzyć, że wartość R_{SD} będzie taka, że już niewielkie zmiany napięć zasilających spowodują przejście napięcia U_k przez próg U_p i związaną z tym zmianę znaku napięcia U_{wy} . Należy więc programować tak, aby rezystancja R_{SD} była zawsze znacznie poniżej rezystancji R_p , tzn. takiej wartości R_{SD} , przy której $U_k = U_p$. Ładunek, jaki trzeba zgromadzić w bramce tranzystora, aby jego rezystancja R_{SD} równała się R_p , nosi nazwę ładunku progowego Q_p . Ładunek ten przy określonym napięciu progowym U_p zależy od napięć zasilających komórkę, charakterystyki wzmocnienia przedwzmacniacza oraz od R_{xy} . Dlatego też wartości ładunku progowego poszczególnych komórek tej samej pamięci mogą się znacznie między sobą różnić.



Rys. 4. Uproszczony schemat zastępczy komórki pamięci w stanie odczytu

Poprawne zaprogramowanie komórki wymaga pewnego przeładowania bramki, które polega na wprowadzeniu do jej obszaru ładunku większego niż progowy. Przeprogramowanie stosuje się w celu zabezpieczenia przed skutkami niekorzystnego działania czynników wpływających na stan naładowania bramki. Podstawowe czynniki wpływające na naładowanie bramki to:

a) Napięcie źródła zasilania

Wpływ napięcia zasilającego PROM jest bardzo nieliniowy i stąd PROM pracujący przy napięciu zasilania mniejszym o 0,5 V od wartości minimalnej może już działać nieprawidłowo. Napięcie zasilania ma również wpływ na wielkość czasu dostępu do pamięci.

b) Neutralizacja ładunku

Proces neutralizacji silnie zależy od temperatury. Na przykład w temperaturze 125°C około 10% nagromadzonego w bramce ładunku ulega neutralizacji, po czym proces ten zanika.

c) Prądy upływu

Niektóre z elektronów zakumulowanych w bramce mogą ją opuścić w postaci prądu upływu, którego wielkość również znacznie zależy od temperatury. W temperaturze 125°C przez okres 10 lat strata ładunku zgromadzonego w bramce może sięgać 10% wartości początkowej. W temperaturze 25°C prąd upływu jest nieznaczny.

d) Temperatura

Temperatura ma decydujący wpływ na wielkość rezystencji R_p , jak również na napięcie progowe U_p . Przy temperaturze 70°C trzeba na przykład zgromadzić w bramce ładunek o 15% większy niż w temperaturze 25°C.

Na podstawie oceny wpływu wymienionych czynników na stan zaprogramowania PROM-u, powszechnie przyjęto [1], że każda programowana komórka powinna mieć ładunek minimalnie o 40% większy od ładunku progowego ($Q=1,4 Q_p$) przy programowaniu w temperaturze 25°C . Żądane przeładowanie komórki uzyskuje się poprzez wydłużenie czasu programowania o wielkość $4t$ (gdzie " t " jest czasem, jaki upłynął od chwili rozpoczęcia programowania do momentu zgromadzenia w bramce ładunku progowego), przez co całkowity czas programowania wyniesie $5t$. Wielkość czasu przeprogramowania ustalono na podstawie charakterystyk programowania "najgorszych" komórek typowej pamięci, które na rys. 3 reprezentuje komórka B.

Ze względu na stosunkowo długi czas programowania komórki (np. dla komórki B z rys. 3 równy 50 ms) nie można stale utrzymywać na niej wysokiego napięcia programującego. Wydzielane podczas programowania ciepło mogłoby uszkodzić zarówno programowaną komórkę, jak i całą pamięć. Dlatego też napięcie programujące podaje się w postaci impulsów (najczęściej 47-woltowych) o czasie trwania $1\text{-}3\text{ ms}$ i współczynniku wypełnienia 20% .

Pamięci PROM FAMOS programowane są słowami, co oznacza, że jednocześnie programuje się wszystkie bity tworzące słowo. Po każdym impulsie programującym następuje odczyt programowanego słowa i porównanie go ze słowem wzorcowym. Operacja ta jest powtarzana aż do momentu, kiedy oba słowa będą identyczne. Ilość impulsów programujących N , konieczna do osiągnięcia równości programowanego słowa ze słowem wzorcowym, jest miarą czasu potrzebnego do zgromadzenia ładunku progowego Q_p w "najgorszej" komórce programowanego słowa. Całe słowo jest następnie przeprogramowywane poprzez podanie $4N$ impulsów programujących, rezultatem czego jest zgromadzenie w bramce "najgorszej" komórki ładunku $1,4 Q_p$. Pozostałe komórki będą oczywiście bardziej przeprogramowane. Maksymalna ilość impulsów programujących jest w większości programatorów ograniczona do 32. Pamięci, w których istnieje komórka nie dająca się zaprogramować, uznawane są za uszkodzone i zostają odrzucone.

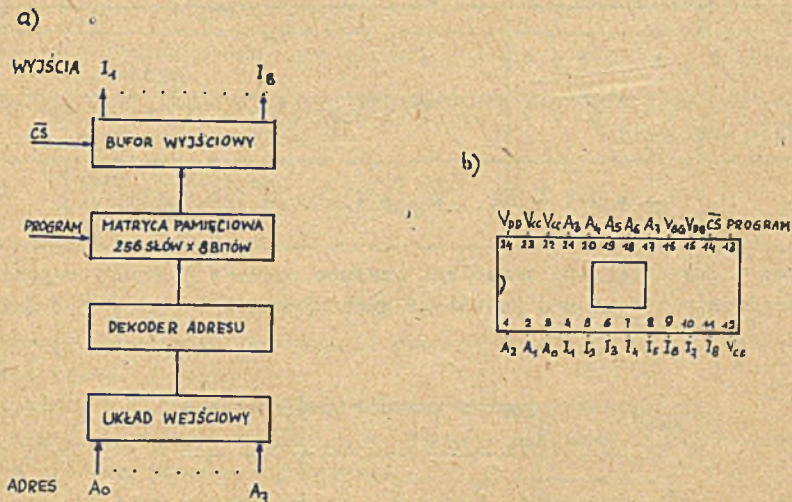
5. Kasowanie zawartości pamięci PROM FAMOS

Kasowanie zawartości pamięci wykonanej w technologii FAMOS odbywa się przez naświetlanie pamięci promieniami ultrafioletowymi o długości fali $253,7\text{ nm}$ poprzez wykonane w obudowie pamięci przezroczyste okienko znajdujące się nad kryształkiem. Zalecane natężenie promieniowania wynosi 6 W/cm^2 . Normalne oświetlenie (sztuczne i słoneczne) nie ma większego wpływu na ładunek zgromadzony w bramce, jakkolwiek zaleca się, aby PROM-y pracowały z zasłoniętymi okienkami. Jeżeli element nie posiada przezroczystego okienka, to do kasowania jego zawartości stosuje się promieniowanie X. Naświetlanie promieniami ultrafioletowymi lub promieniami X jest przyczyną przepływu prądu fotoelektrycznego z bramki do podłoża, który powoduje usunię-

cie ładunku z bramki. Całkowite rozładowanie bramki uzyskuje się po czasie większym od 20 minut.

6. Programowana pamięć stała INTEL 1702A

Pamięć stała INTEL 1702A jest pamięcią o organizacji 256 słów 8-bitowych/zamkniętą w obudowie posiadającej 24 wyprowadzenia.



Rys. 5

a - schemat blokowy pamięci 1702A, b - widok zewnętrzny pamięci 1702A

Na rys. 5a przedstawiono schemat blokowy pamięci 1702A, a na rys. 5b jej widok zewnętrzny wraz z organizacją wyprowadzeń. Poszczególne wyprowadzenia mają następujące znaczenie:

- I_1, I_8 są trójstanowymi wyjściami informacyjnymi, które mogą przyjmować stany: niski (L), wysoki (H) oraz stan wysokiej impedancji (WI). Podczas procesu programowania wyjścia informacyjne pełnią rolę wejść danych,
- A_0+A_7 wejścia adresowe,
- \overline{CS} sygnał selekcji,
- PROGRAM sygnał programujący pamięć,
- $V_{CC}, V_{DD}, V_{GG}, V_{BB}$ sygnały zasilające pamięć (stałe podczas odczytu, impulsowe podczas programowania).

W tabeli 1 przedstawiono napięcia pojawiające się na wejściach zasilających i sterujących pamięci podczas operacji czytania i programowania.

Tabela 1

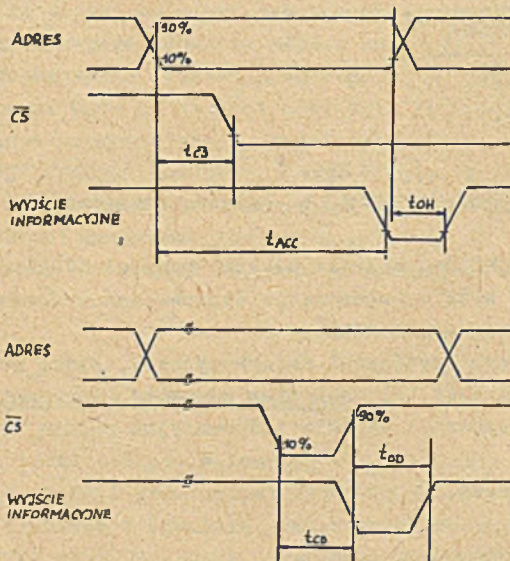
Nr wyprowadzenia wraz z oznaczeniem sygnału	12 (V_{CC})	13 (PROGRAM)	14 (\overline{CS})	15 (V_{BB})	16 (V_{GG})	22 (V_{CC})	23 (V_{CC})	24 V_{DD}
Czytanie	V_{CC}	V_{CC}	Masa	V_{CC}	V_{GG}	V_{CC}	V_{CC}	V_{DD}
Programowanie	Masa	IMPULS PROGRAMUJACY	Masa	V_{BB}	IMPULS V_{GG}	Masa	Masa	IMPULS V_{DD}

Operacja czytania

Podczas operacji czytania poszczególne napięcia zasilające pamięć mają następujące wartości:

$$V_{CC} = 5 \text{ V} \pm 5\%, \quad V_{DD} = V_{GG} = -9 \text{ V} \pm 5\%$$

Na rys. 6 przedstawiono przebiegi czasowe podczas operacji czytania [2]. Czasy narastania i opadania poszczególnych sygnałów powinny być mniejsze od 50 ns.



Rys. 6. Przebiegi czasowe sygnałów na wejściach PROM-u podczas operacji czytania

Tabela 2 zawiera wartości przedziałów czasowych zaznaczonych na rys. 6, których nie można przekroczyć, aby odczyt był poprawny.

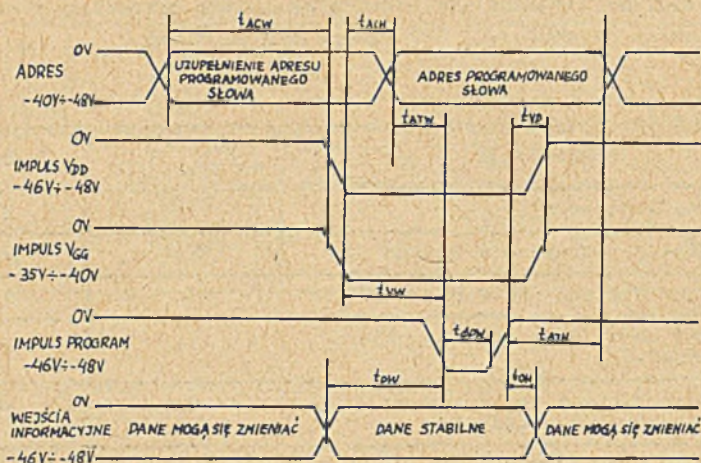
Tabela 2

Symbol	Określenie	Wartość maksymalna	Jednostka
f	Częstotliwość	1	MHz
t_{OH}	Czas przetrzymania danych	0,1	μs
t_{ACC}	Czas dostępu do danych	1	μs
$t_{\overline{CS}}$	Max czas opóźnienia sygnału selekcji	0,1	μs
t_{CO}	Czas dostępu do danych po zmianie sygnału selekcji	0,9	μs
t_{OD}	Czas przetrzymania danych na wyjściu przy zmianie sygnału selekcji	0,3	μs

Operacja programowania

Operacja programowania (rys. 7) pamięci 1702A odbywa się według następujących zasad:

1. Wszystkie 8 bitów słowa programowanych jest jednocześnie.
2. Po każdym impulsie programującym testuje się stan zaprogramowania wszystkich bitów programowanego słowa.
3. Maksymalny czas programowania wszystkich 2048 bitów wynosi 2 min.
4. Parametry impulsów sterujących programowaniem:
 - Impuls programujący PROGRAM
amplituda $-46 V \pm -48 V$
czas trwania $2 ms \pm 3 ms$
współczynnik wypełnienia 20%.
 - Impuls V_{DD}
amplituda $-46 V \pm -48V$
czas trwania determinują odcinki czasowe t_{VW} i t_{VD} (rys. 8 i tabela 3)
 - Impuls V_{GG}
amplituda $-35 V \pm -40 V$
czas trwania determinują odcinki czasowe t_{VW} i t_{VD} .
5. Przy programowaniu logicznej jedynek wymagane jest podanie na wejścia informacyjne napięcia $-47 V$, przy programowaniu logicznego zera napięcia $0V$. Dane powinny pojawić się na wejściach informacyjnych o czas t_{DW} przed impulsem programującym i muszą być przetrzymane przez czas t_{DH} po jego zakończeniu.



Rys. 7. Przebiegi czasowe sygnałów na wejściach PROM-u podczas operacji programowania

Tabela 3

Symbol	Określenie	Wartość			Jednostka
		min	typowa	max	
t_{OPW}	Czas trwania impulsu programującego		2	3	ms
t_{DW}	Czas wyprzedzenia danych	25			μs
t_{DH}	Czas przetrzymania danych	10			μs
t_{VW}	Czas wyprzedzenia impulsu programującego przez impulsy V_{GG} i V_{DD}	100			μs
t_{VD}	Czas przetrzymania impulsów V_{GG} i V_{DD} po zakończeniu impulsu programującego	10		100	μs
t_{ACW}	Czas przetrzymania uzupełnienia adresu po zakończeniu impulsów V_{DD} i V_{GG}	25			μs
t_{ACH}	Czas wyprzedzenia impulsów V_{GG} i V_{DD} przez uzupełnienie adresu	25			μs
t_{ATW}	Czas wyprzedzenia impulsu programującego przez adres	10			μs
t_{ATH}	Czas przetrzymania adresu po zakończeniu impulsu programującego	10			μs

6. Stanowi logicznej jedyńki w słowie adresowym odpowiada podczas programowania napięcie -47 V na wejściu adresowym pamięci, stanowi logicznego zera napięcie 0 V . Przedtem konieczne jest jednak podanie na wejścia adresowe pamięci uzupełnienia do jedyńki adresu programowanego słowa przez czas t_{ACH} przed impulsem V_{DD} .

Na rys. 7 przedstawiono przebiegi czasowe sygnałów na wejściach adresowych, informacyjnych oraz zasilających podczas programowania.

Tabela 3 zawiera określenie czasów zaznaczonych na rys. 7 wraz z podaniem ich wartości. Czasy narastania i opadania impulsów przedstawionych na rys. 7 powinny być mniejsze od $1\ \mu\text{s}$.

Pozostałe ważniejsze dane pamięci 1702A:

- pamięć jest przeznaczona do współpracy z układami TTL i DTL,
- pojemność układów wejściowych i wyjściowych wynosi 15 pF ,
- natężenie światła podczas wymazywania informacji wynosi 6 W/cm^2 , a jego długość $253,7\text{ nm}$,
- temperatura pracy $0^\circ\text{C} \pm 70^\circ\text{C}$.

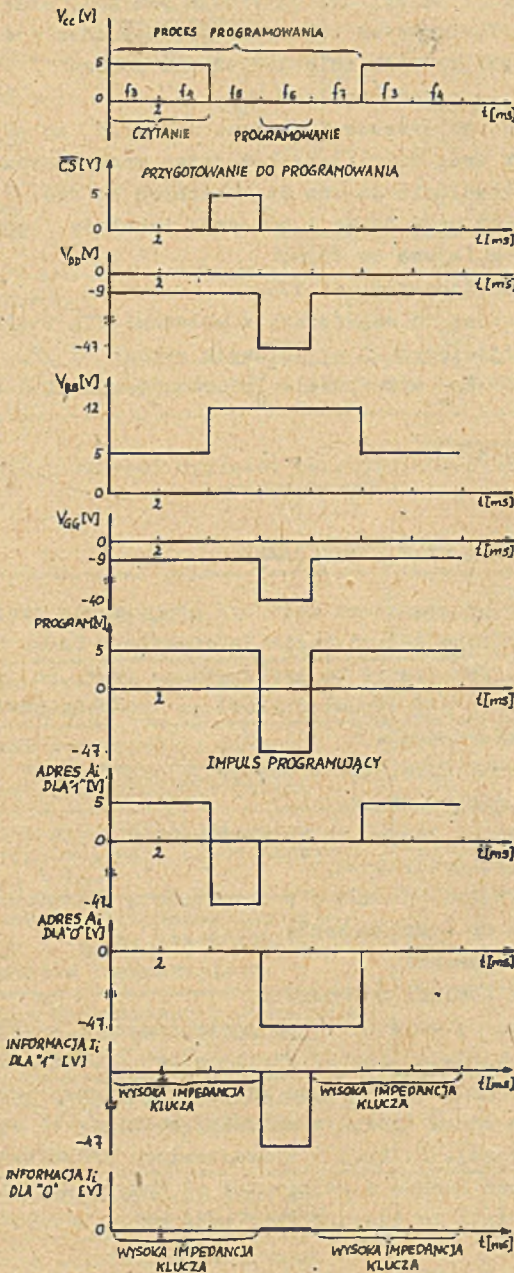
7. Programator pamięci PROM FAMOS 1702A

Przykładem prostego programatora może być programator oznaczony symbolem HM1, który zrealizowano w Instytucie Informatyki Czasu Rzeczywistego Politechniki Śląskiej. Urządzenie to przeznaczone jest do programowania pamięci INTEL 1702A (oraz jej odpowiedników) za pośrednictwem klawiatury lub czytnika taśmy perforowanej.

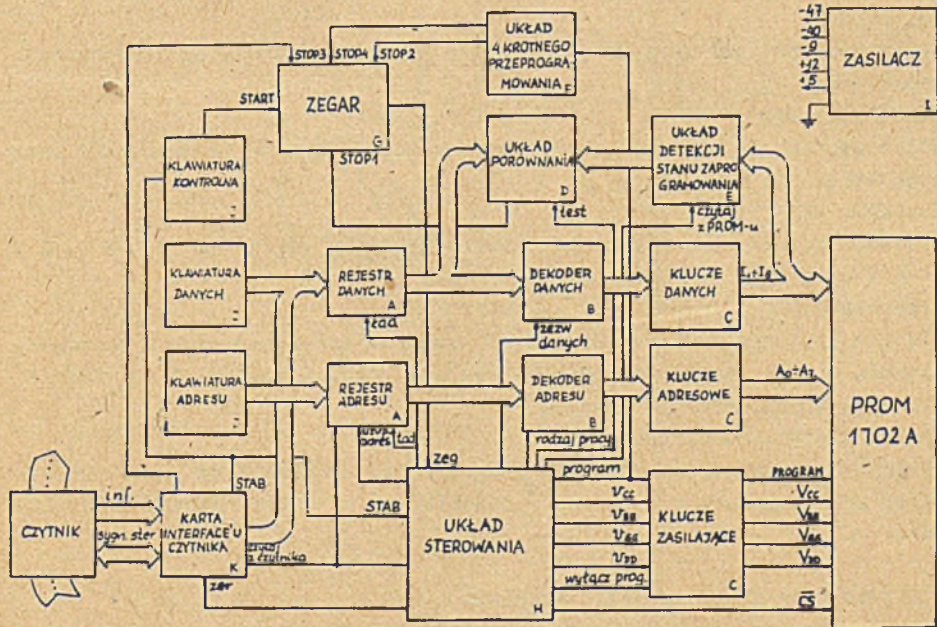
Praca programatora HM1 przebiega w następujących fazach:

- f_1 - stabilizacja programatora,
- f_2 - pobranie adresu i danych do rejestru,
- f_3, f_4 - czytanie zawartości PROM-u,
- f_5 - porównanie zawartości PROM-u z zawartością rejestru danych i przygotowanie do programowania,
- f_6 - programowanie pamięci,
- f_7 - przygotowanie pamięci do odczytu.

Proces programowania sprowadza się do konieczności wygenerowania na wejściach pamięci odpowiednich napięć zasilających zależnych od fazy, w jakiej znajduje się programator oraz od napięć reprezentujących programowane słowo, jak i jego adres (również zmieniających się w poszczególnych fazach pracy programatora). Na rys. 8 przedstawiono przebiegi sygnałów na poszczególnych wejściach PROM-u w zależności od fazy, w jakiej znajduje się programator (pominięto tu jedynie fazy wstępne f_1 i f_2).



Rys. 8. Przebiegi sygnałów na wejściach PROM-u w zależności od fazy pracy programatora



Rys. 9. Schemat blokowy programatora HM1

Rysunek 9 przedstawia schemat blokowy programatora HM1. Poszczególne bloki tego urządzenia pełnią następującą rolę:

- Rejestr danych i rejestr adresu (A)
Rejestr danych służy do przechowywania programowanego słowa. Adres programowanego słowa zawarty jest w rejestrze adresu.
- Dekoder danych i dekodek adresu (B)
Dekoder danych steruje przełączaniem kluczy danych, dekodek adresu steruje przełączaniem kluczy adresowych.
- Klucze danych, klucze adresowe i klucze zasilające (C)
Układ kluczy wprowadza na wejścia PROM-u napięcia reprezentujące programowane słowo, jego adres oraz napięcia zasilające zgodnie z przebiegami przedstawionymi na rys. 8.
- Układ porównania (D)
Układ porównania służy do porównania programowanego słowa (które pełni rolę wzorca) ze słowem odczytanym z PROM-u w celu sprawdzenia poprawności zaprogramowania.
- Układ detekcji stanu zaprogramowania (E)
Detektor stanu zaprogramowania PROM-u jest układem przeznaczonym do wykrycia momentu, w którym komórkę można uznać za zaprogramowaną.

- Układ czterokrotnego przeprogramowania (F)
Układ ten zapewnia odpowiednie (czterokrotne) przeprogramowanie PROM-u.
- Zegar (G)
Zegar przeznaczony jest do generowania sygnału taktującego pracę programatora.
- Układ sterowania (H)
Układ sterowania steruje pracą poszczególnych bloków programatora oraz organizuje ich współpracę.
- Zasilacz (I)
Zasilacz pozwala uzyskać stabilizowane napięcia -47 V , -40 V , -9 V , -12 V i -5 V konieczne do pracy programatora.
- Klawiatura (J)
Klawiatura służy do wprowadzania adresu oraz przeznaczonych do zaprogramowania danych, umożliwia również obsługę programatora (stabilizację, start, wybranie rodzaju pracy).
- Karta interface'u czytelnika (K)
Układ ten organizuje współpracę programatora z czytnikiem oraz pośredniczy przy wprowadzaniu danych z czytnika do rejestru danych programatora.

7.1. Rejestr danych i rejestr adresu

Rejestr danych jest prostym ośmiobitowym rejestrem, którego ładowanie może odbywać się z klawiatury za pomocą sygnału "ład" oraz z czytnika taśmy perforowanej za pomocą sygnału "czytaj z czytnika".

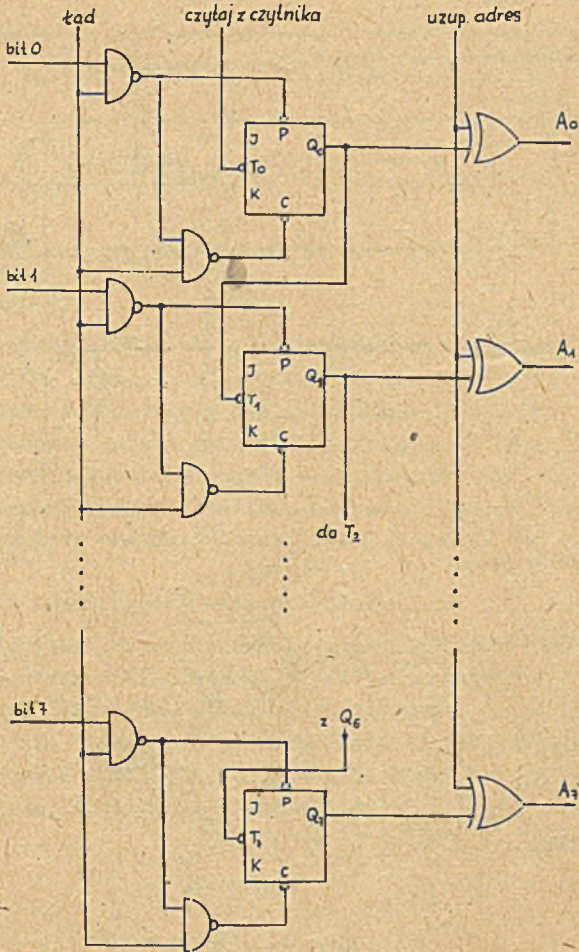
Rejestr adresu stanowi osiem przerzutników J-K połączonych w układ binarnego licznika asynchronicznego. Wprowadzanie adresu do rejestru może odbywać się dwojako:

- przy programowaniu z klawiatury adres wpisywany jest za pomocą sygnału "ład" poprzez wejścia PRESET i CLEAR poszczególnych przerzutników,
- przy programowaniu z czytnika kolejne adresy generowane są wewnątrz rejestru poprzez zliczanie impulsów "czytaj z czytnika".

Na wyjściu rejestru znajduje się układ ośmiu bramek EXCLUSIVE-OR, na wyjściu których pojawia się adres programowanego słowa (jeżeli sygnał "uzup. adres" jest w stanie 0) lub jego uzupełnienie do jedynki (jeżeli sygnał "uzup. adres" jest w stanie 1). Rys. 10 przedstawia rejestr adresu wraz z układem uzupełnienia adresu.

7.2. Dekoder danych i dekodek adresu

Osiem bitów rejestru danych wyprowadzone zostało na dekodek danych, który w fazie programowania zamyka klucze danych przyłączające na wejścia danych PROM-u wysokie napięcie (-47 V) przy programowaniu jedynki i niskie (0 V) przy programowaniu zera. W czasie odczytu zawartości PROM-u dekodek danych otwiera klucze, powodując tym samym pojawienie się wysokiej impe-



Rys. 10. Rejestr adresu wraz z układem uzupełnienia adresu

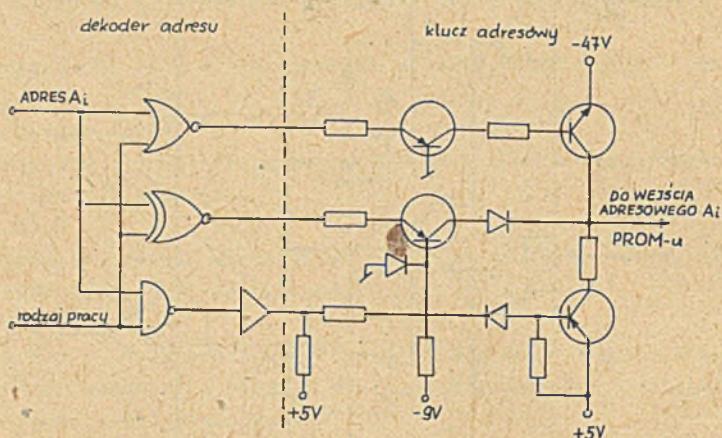
dancji na wejściach danych PROM-u (impedancja widziana z wejście PROM-u w kierunku klucza).

Tablicę wzbudzeń dla wejść danych zawiera tabela 4.

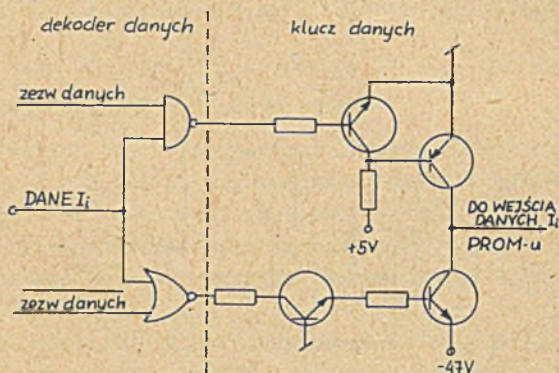
Tabela 4

Faza pracy	Dane	Zezwolenie danych	Napięcie na wejściu danych PROM-u
Czytanie	1 0	0 0	Klucz otwarty (wysoka impedancja)
Programowanie	1 0	1 1	-47 V 0 V

a)



b)



Rys. 11

a - dekoder adresu wraz z odpowiadającym mu kluczem adresowym, b - dekoder danych wraz z odpowiadającym mu kluczem danych

Na rys. 11b przedstawiono dekoder sterujący przełączaniem napięć na wejściu danych wraz z odpowiadającym mu kluczem danych.

Osiem bitów rejestru adresu wyprowadzonych jest do dekodera adresu, który zależnie od fazy programowania zamyka lub otwiera klucze adresowe przyłączające na wejścia adresowe programowanego elementu właściwe w danej fazie napięcie. W tabeli 5 przedstawiono tablicę wzbudzeń wejścia adresowego.

Dekoder sterujący przełączaniem napięć na wejście adresowe wraz z odpowiadającym mu kluczem adresowym przedstawiono na rys. 11a.

Tabela 5

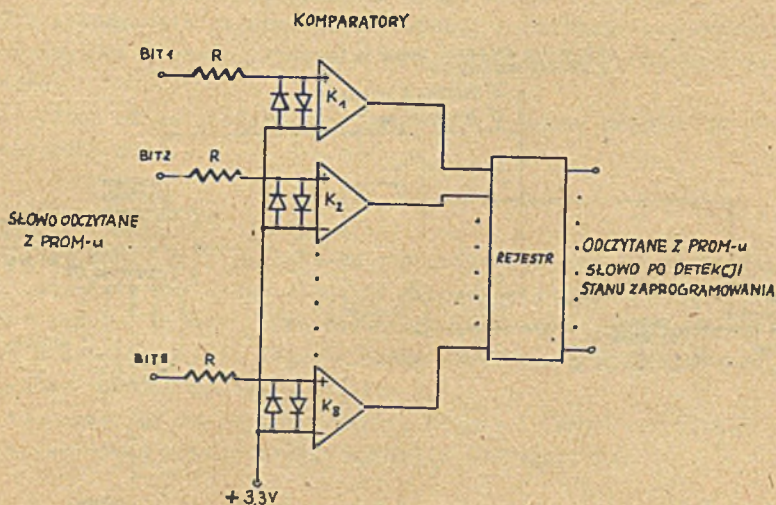
Faza pracy	Adres	Rodzaj pracy	Napięcie na wejściu adresowym PROM-u odpowiadające danej fazie pracy
Czytanie	0	1	0 V
	1	1	5 V
Programowanie	1	0	0 V
	0	0	- 47 V

7.3. Układ kluczy

Wszystkie sygnały sterujące pracą programatora HM1 generowane są przez układy TTL. Poziomy sygnałów koniecznych do zaprogramowania PROM-u i odczytania jego zawartości różnią się zasadniczo od poziomu TTL (rys. 8), skąd wynika konieczność zastosowania kluczy tranzystorowych, których zadaniem jest przekształcenie sygnałów w standardzie TTL na sygnały o poziomach wymaganych przez programowany element. Czas narastania i opadania sygnałów na wyjściach poszczególnych kluczy powinien być mniejszy od 1 μ s.

7.4. Detektor stanu zaprogramowania PROM-u

Do testowania stanu zaprogramowania poszczególnych komórek PROM-u służy detektor zaprogramowania PROM-u. Układ ten składa się z ośmiu komparatorów analogowych i rejestru stanu zaprogramowania pamięci. Na komparatorach dokonuje się w fazie odczytu sprawdzenia, czy napięcie na wyjściach informacyjnych odpowiadających komórkom, do których wpisywano podczas programowania jedynek, przekroczyło wartość 3,3 V. Przekroczenie 3,3 V na

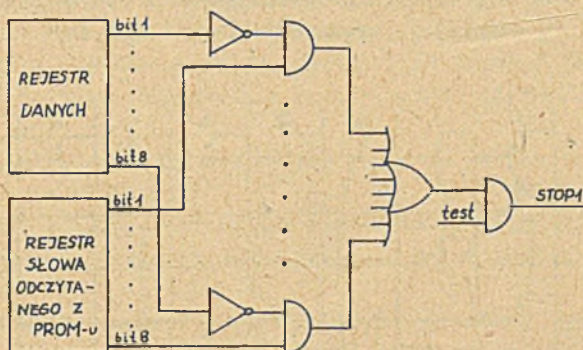


Rys. 12. Detektor stanu zaprogramowania PROM-u

7.6. Zegar

Zegar programatora HM1 jest generatorem impulsów prostokątnych o czasie trwania 1 ms i współczynniku wypełnienia 0,5. Uruchomienie zegara odbywa się za pomocą przycisku "START", natomiast jego zatrzymanie może nastąpić w czterech przypadkach:

STOP1 - informuje, że w komórce, w której ma zostać zaprogramowane logiczne zero, znajduje się już zapisana logiczna jedynka. Zaprogramowanie logicznego zera w tej komórce jest możliwe dopiero po wymazaniu całej zawartości PROM-u. Na rys. 14 przedstawiono układ generacji sygnału STOP1.



Rys. 14. Układ generacji sygnału STOP1

STOP2 - przepełnienie licznika układu porównania. Przepełnienie licznika ma miejsce, kiedy dane słowo nie zostanie zaprogramowane, co może być spowodowane uszkodzeniem programowanego słowa (rys. 13).

STOP3 - zakończenie transmisji informacji z czytnika do rejestru danych programatora i zaprogramowanie ostatniego słowa wprowadzonego do tego rejestru.

STOP4 - dane słowo zostało zaprogramowane i można przejść do programowania kolejnych słów.

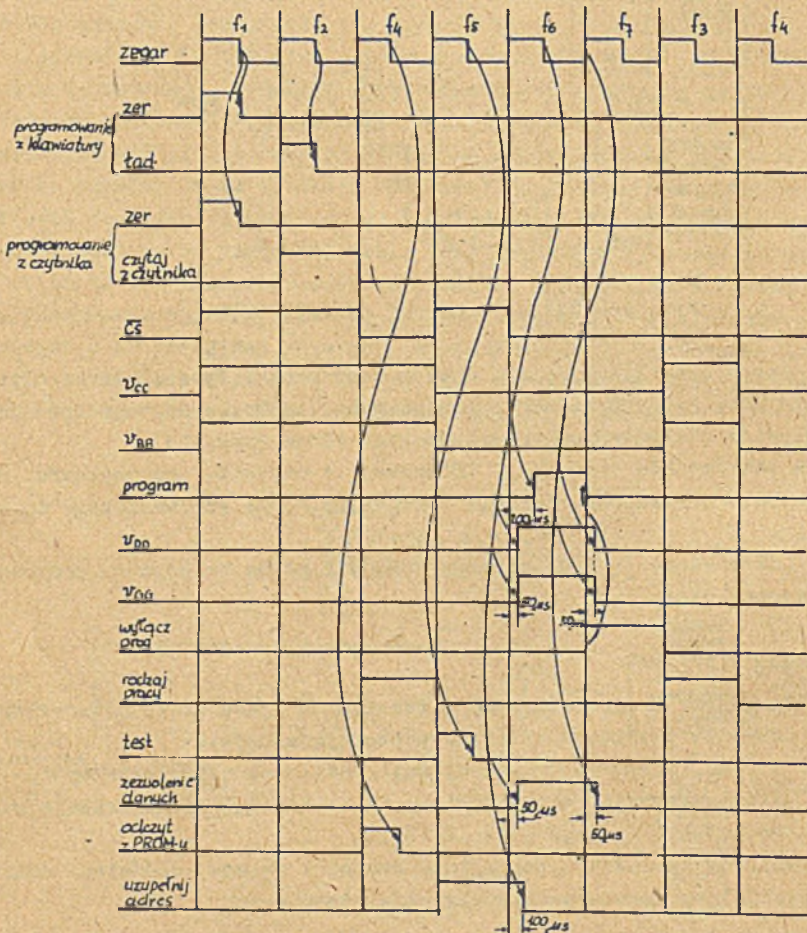
7.7. Układ sterowania

Układ sterowania programatora HM 1 rozwiązano jako układ mikroprogramowany. W układzie tym można wyróżnić następujące bloki:

- rejester adresu, w którym przechowywany jest adres mikroinstrukcji,
- dekodek mikroinstrukcji, który wybiera odpowiednią mikroinstrukcję zgodnie z kodem podanym na jego wejściu,
- układ kombinacyjny zawierający układy bramek i układy opóźnień czasowych zapewniające wygenerowanie sygnałów sterujących,

matrycę diodową, której linie wierszy reprezentują poszczególne fazy pracy programatora, a linie kolumn mikro sygnały sterujące. Wygenerowanie w danej fazie wszystkich mikro sygnałów potrzebnych do wykonania określonej czynności składa się na mikroinstrukcję.

Programator HM1 pracuje w siedmiu fazach, przy czym fazy f_1 i f_2 są fazami wstępnymi i służą jedynie do przygotowania programatora do programowania. W fazie f_1 zerowany jest licznik impulsów programujących, a w fazie f_2 zostaje wpisane programowane słowo do rejestru danych i jego adres do rejestru adresu. Podczas programowania programator pracuje więc tylko w pięciu fazach od f_3 do f_7 . Właściwe programowanie pamięci związane jest z pojawieniem się impulsu programującego i odbywa się w fazie f_6 . Fazy f_3 , f_4 , f_5 i f_7 zapewniają właściwy odstęp pomiędzy impulsami pro-

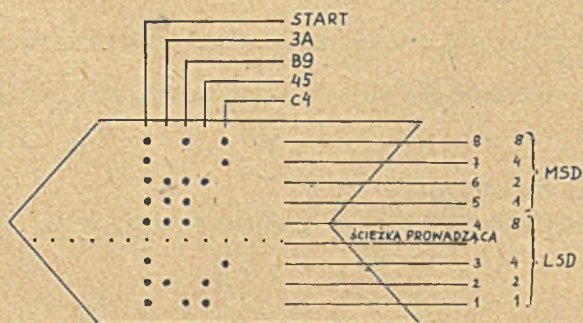


Rys. 15. Przebiegi czasowe sygnałów sterujących pracą programatora HM1

gramującymi (impulsy programujące mają współczynnik wypełnienia 20%) i wykorzystane są do: przygotowania PROM-u do odczytu (f_7), odczytania zawartości ośmiu aktualnie programowanych komórek (f_3, f_4) oraz do porównania zawartości rejestru stanu zaprogramowania PROM-u z zawartością rejestru danych (f_5). Zatrzymanie pracy programatora następuje po wyzerowaniu licznika impulsów programujących, co powoduje wygenerowanie sygnału STOP 4 i ponowne przejście układu do fazy wstępnej f_1 . Na rys. 15 przedstawiono przebiegi czasowe sygnałów sterujących pracą programatora HM1.

Poszczególne sygnały mają następujące znaczenia:

- "zer" - sygnał zerujący licznik impulsów programujących i ustawiający przerzutnik karty interface'u czytelnika,
- "ład" - sygnał ładowania rejestru danych i rejestru adresu,
- "wyłącz prog" - sygnał powodujący zakończenie impulsu programującego
- "rodzaj pracy"-sygnał określający rodzaj pracy (czytanie lub programowanie PROM-u),
- "test" - sygnał porównania zawartości rejestru danych i rejestru stanu zaprogramowania PROM-u,
- "zezw. danych" - sygnał powodujący wprowadzenie programowanego słowa na wejścia kluczy przełączających napięcia na wejściach informacyjnych PROM-u,
- "czytaj z PROM-u" - sygnał powodujący wpisanie słowa znajdującego się na wyjściach komparatorów układu detektora stanu zaprogramowania PROM-u do rejestru stanu zaprogramowania PROM-u,
- "uzup. adres" - sygnał uzupełnienia adresu (uzupełnienie do "1").



Rys. 16. Format taśmy perforowanej przy programowaniu z czytelnika

8. Format taśmy perforowanej przy programowaniu z czytelnika

Format taśmy perforowanej przedstawiono na rys. 16. W formacie tym [•] jeden rząd na taśmie perforowanej reprezentuje osiem bitów programowanego słowa. Programowaną informację poprzedza znak startu, którym jest

rządek zawierający same dziurki. Przy zapisie szesnastkowym programowanej informacji należy części bardziej znaczącej zapisanej cyfry (MSD) przypisać prawą stronę taśmy perforowanej (w stosunku do ścieżki prowadzącej). Programowanie pamięci z czytnika rozpoczyna się od słowa o adresie ustalonym na klawiaturze, po czym programowane są słowa o kolejnych adresach.

9. Uwagi końcowe

W niniejszym artykule omówiono zasady programowania pamięci typu FAMOS i przedstawiono prosty programator, który umożliwia programowanie pamięci określonego typu jedynie z klawiatury oraz z czytnika taśmy perforowanej. Programator ten nie nadaje się do programowania pamięci o innej organizacji wyprowadzeń i innych napięciach zasilających. Należy jednak podkreślić, że zasada programowania wszystkich pamięci typu FAMOS jest taka sama i stąd większość układów programatora HM1 może być przystosowana do programowania pamięci o innej organizacji, większej pojemności czy też innych napięciach zasilających.

LITERATURA

- [1] PROM User's Guide, Prolog Corporation, Monterey, May 1977.
- [2] Memory Design Handbook, Intel Corporation, Santa Clara 1977.
- [3] Data Catalog, Intel Corporation, Santa Clara, 1977.
- [4] Hnatek E.: Current Semiconductor Memories. Computer Design pp.115-126, April 1978.
- [5] Condensed Catalog LSI, Microcomputer Memories CCD, Fairchild, Mountain View, May 1978.
- [6] Wagner F.: Projektowanie urządzeń cyfrowych. WNT, Warszawa 1978.

ПРОГРАММИРОВАНИЕ МОС-ЭПРОМ-У - ЭЛЕКТРИЧЕСКИ РЕПРОГРАММИРОВАННОГО ЗАПОМИНАЮЩЕГО УСТРОЙСТВА

Р е з ю м е

Основным элементом микропрограммированных цифровых оборудований является программированное полупроводниковое запоминающее устройство ПРОМ. Самыми популярными являются электрически программированные ПРОМ-ы - ЭПРОМ-ы.

В связи с тем возникает вопрос простого, надёжного и дешёвого программирования этих элементов.

Автор рассматривает технику программирования ЭПРОМ-ов типа FAMOS и конструкцию оборудования для программирования ЭПРОМ-у 1702 А.

PROGRAMMING THE MOS-EPROMs

ELECTRICALLY REPROGRAMMABLE READ-ONLY MEMORIES

S u m m a r y

The basic device used in microprogrammable digital equipment is programmable semiconductor read-only memory PROM. The most popular are electrically programmable and reprogrammable PROMs-EPROMs, so a problem of easy, quick, reliable and low cost programming of these devices is essential. The article describes the programming technique of EPROMs FAMOS type and presents the design of programmer for programming "industry standard" EPROM 1702A.