

Marek CHMURA

PAMIĘĆ SKOJARZENIOWA

Streszczenie. W artykule zostały przedstawione dwa rozwiązania pamięci adresowanych zawartością. Pierwsze, to powszechnie przyjęte w różnych układach półprzewodnikowych tworzących pamięć CAM. Drugie przedstawia nową organizację bardzo szybkiej pamięci skojarzeniowej. Oba rozwiązania zostały przedstawione pod kątem zastosowania, w celu szukania elementu ekstremalnego spośród zapamiętanych w pamięci. Szukanie wykonywane jest w paru krokach na drodze analizy kolejnych bitów liczby binarnej zaczynając od lewego po jednym na raz. Algorytm poszukiwania polega na znalezieniu zbioru elementów predystynowanych do wystąpienia w roli elementu ekstremalnego, spośród których w każdym kroku usuwa się podzbiór aż pozostaną wyłącznie najmniejsze lub największe.

1. WSTĘP

Pamięcią skojarzeniową nazywana jest pamięć o organizacji rozszerzającej jej standardowe funkcje w najprostszej postaci o operację wyszukiwania adresów komórek, zawierających informację równą zadanemu argumentowi¹⁾. Pamięć o takich możliwościach jest określana w literaturze również jako pamięć asocjacyjna lub pamięć adresowana zawartością w skrócie CAM (Content Adressable Memory) [1], [3].

Dalsze rozszerzenie funkcji pamięci skojarzeniowej można uzyskać przez przystosowanie jej do skojarzeniowego przetwarzania danych (ADP - Associative Data - Processing), czyli do bardziej - niż proces badania zgodności - złożonych operacji określania i lokalizacji grup danych, np.: poszukiwanie wartości większej, mniejszej lub równej zadanej, następnej większej lub mniejszej, zawartej pomiędzy wartościami granicznymi, maksymalnej lub minimalnej itp. [2], [3]. Sprzętowa realizacja tych operacji jest najczęściej wykonywana na drodze mikroprogramowania działań nad zawartością pamięci CAM [1], [3].

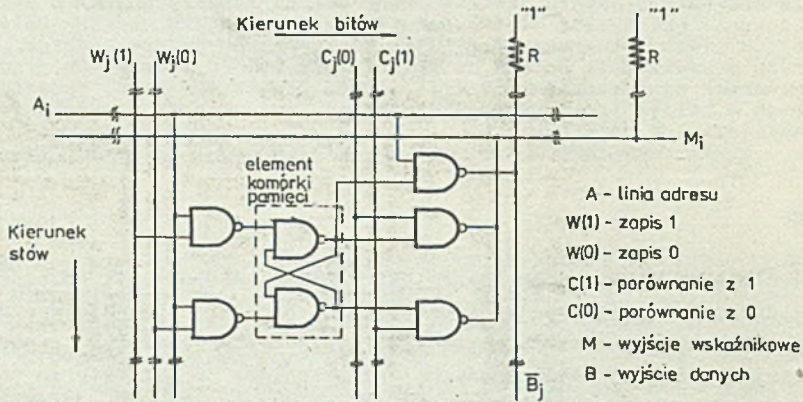
W poniższym artykule przedstawione są rozważania dotyczące pamięci skojarzeniowej poszukującej w zadanym zbiorze elementów ekstremalnych. Organizacja pamięci o mikroprogramowanym sterowaniu operacjami poszukiwania została przedstawiona w rozdziale drugim. Takie rozwiązanie nie zawsze jest zadowalające szczególnie wtedy, gdy zależy nam na bardzo szybkiej odpowiedzi. Wymaganie to spełnia pamięć skojarzeniowa o sterowaniu ukła-

¹⁾ Np. element firmy Signetics GX8 10155 lub 8220.

dowym nazywanym też logiką "zaszytą" prezentowana w rozdziale trzecim. Rozwiązanie to cechuje nie tylko duża szybkość ale również możliwość poszukiwania ekstremalnej wśród liczb o różnych znakach, przedstawionych w zapisie uzupełnienie do 2.

2. PAMIĘĆ SKOJARZENIOWA POSZUKUJĄCA ELEMENTY EKSTREMALNE STEROWANA MIKROPROGRAMOWO

Powszechnie przyjętą w praktycznych zastosowaniach strukturę logiczną komórki pamięci CAM przedstawia rys. 2.1. W strukturze komórki linia A_j



Rys. 2.1. Struktura jednobitowej komórki pamięci CAM stosowanej w spotykanych rozwiązaniach

odgrywa rolę linii adresowej, wybierającej w pamięci słowo zarówno w celu zapisu, jak i odczytu. Przy zapisie informacji wykorzystywane są ponadto linie $W_j(0)$ i $W_j(1)$ wspólne dla danego bitu wszystkich słów pamięci. Stan $W_j(0) = W_j(1) = 0$ oznacza nieaktywność linii zapisu. Zapis jedynki wymaga $W_j(1) = 1$ oraz $W_j(0) = 0$, natomiast zapis zera odwrotnej recelacji.

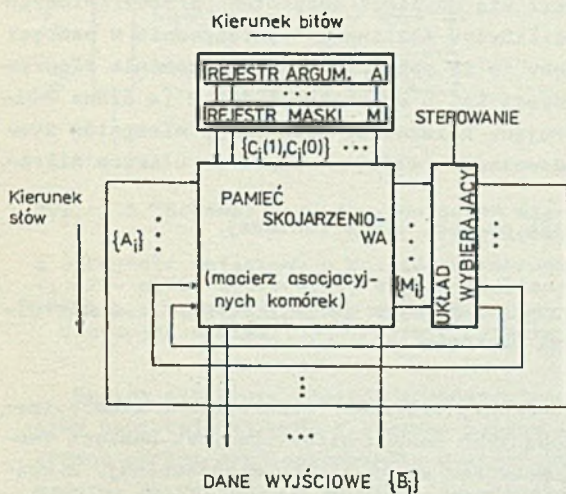
Wybranie słowa linia A_j umożliwia odczyt zanegowanej postaci bitu j-tej pozycji na linii \bar{B}_j .

Z operacją poszukiwania zadanej liczby związana jest linia wyjścia wskaźnikowego M_j oraz linie sterujące $C_j(0)$ i $C_j(1)$. Linia M_j jest wspólna dla wszystkich bitów i-tego słowa a linie $C_j(0)$ i $C_j(1)$ są wspólne dla j-tego bitu wszystkich słów pamięci. Stan linii $C_j(0) = 1$ a $C_j(1) = 0$ ma miejsce wówczas, gdy pytamy czy wśród wszystkich słów zapisanych w pamięci są takie, których j-ty bit ma wartość równą 0. Dla słów, w których ta zgodność wystąpi stan wyjścia zachowa wartość jeden a dla pozostałych,

gdzie zgodności nie ma przyjmie stan zero. Dla poszukiwania 1 linie C_j muszą zmienić się stanami. Maskowanie danego bitu wymusza stan $C_j(0) = C_j(1) = 0$.

Przedstawiona komórka stanowi element konstrukcyjny pamięci CAM (rys. 2.2). Liczbę komórek tworzących pamięć określa długość słowa i jej pojemność. Poszukiwany w pamięci argument jest przechowywany w rejestrze A. Do rejestru M jest wpisywane słowo maskujące. Linie sterujące $C_j(0)$ i $C_j(1)$ są określane na podstawie odpowiednich bitów słowa argumentu i maski.

Poszukiwaną cechę może posiadać wiele słów w wyniku czego otrzymamy wieloelementowy zbiór aktywnych sygnałów M_1 . Bezkolizyjny odczyt poszukiwanych słów umożliwia układ wybierający, który otrzymuje na wejściu stany wszystkich linii M_1 a na wyjściu spośród aktywnych podaje w danym momencie tylko jeden. W wyniku tego otrzymamy na liniach B_j zanegowaną postać poszukiwanego słowa o najmniejszym numerze linii M_1 . Po odczycie słowa odpowiadająca mu linia M_1 zostaje wyzerowana i z pozostałych aktywnych układ wybierający podaje na wyjście znowu tę o najniższym numerze. Operacja ta jest powtarzana tak długo, aż wszystkie słowa zostaną odczytane.



Rys. 2.2. Organizacja pamięci CAM

Przedstawiona na rys. 2.2 pamięć CAM może być wykorzystana do poszukiwania liczb ekstremalnych. Przyjmując, że w pamięci będą zapisywane wyłącznie liczby dodatnie przedstawione w skali rejestru algorytm poszukiwania liczby maksymalnej w tak określonym zbiorze można przedstawić następująco:

1^o do rejestru argumentu na pierwszą pozycję wpisujemy 1. W rejestrze M wszystkie pozycje na prawo od pierwszej maskujemy. Poszukiwane słowo ma postać 1XXX...XX - gdzie X oznacza pozycję maskowaną.

2^o poszukujemy w pamięci słowa równego argumentowi:

- jeżeli w wyniku tego stan wszystkich linii M_1 stanie się równy zero, to przechodzimy do pkt. 4,
- jeżeli w wyniku tego stan tylko jednej linii pozostanie równy 1, to poszukiwanie jest zakończone,

- jeżeli w wyniku tego stan więcej niż jednej linii M_1 pozostanie równy 1, to przechodzimy do pkt. 3.

3^o pozostawiamy 1 na danej pozycji w rejestrze argumentu i jeżeli testowaliśmy ostatni bit słowa, to poszukiwanie jest zakończone a jeżeli nie, to w następnym przejściu wpisujemy 1 na kolejną pozycję rejestru argumentu a pozostałe od niego na prawo bity pozostają nadal maskowane i przechodzimy do pkt. 2.

4^o wpisujemy 0 do danej pozycji rejestru argumentu i jeżeli był to ostatni bit słowa kończymy poszukiwanie a jeżeli nie, to wpisujemy 1 do następnej pozycji rejestru, przy czym wszystkie na prawo od niej pozostawiamy zamaskowane i przechodzimy do pkt. 2.

W najgorszym przypadku dla pamięci o n bitowej długości słowa trzeba wykonać n przejść, w wyniku których otrzymamy nie tylko zbiór linii M_1 wskazujących lokalizację maksymalnych słów ale i jego postać zapisaną w rejestrze argumentu.

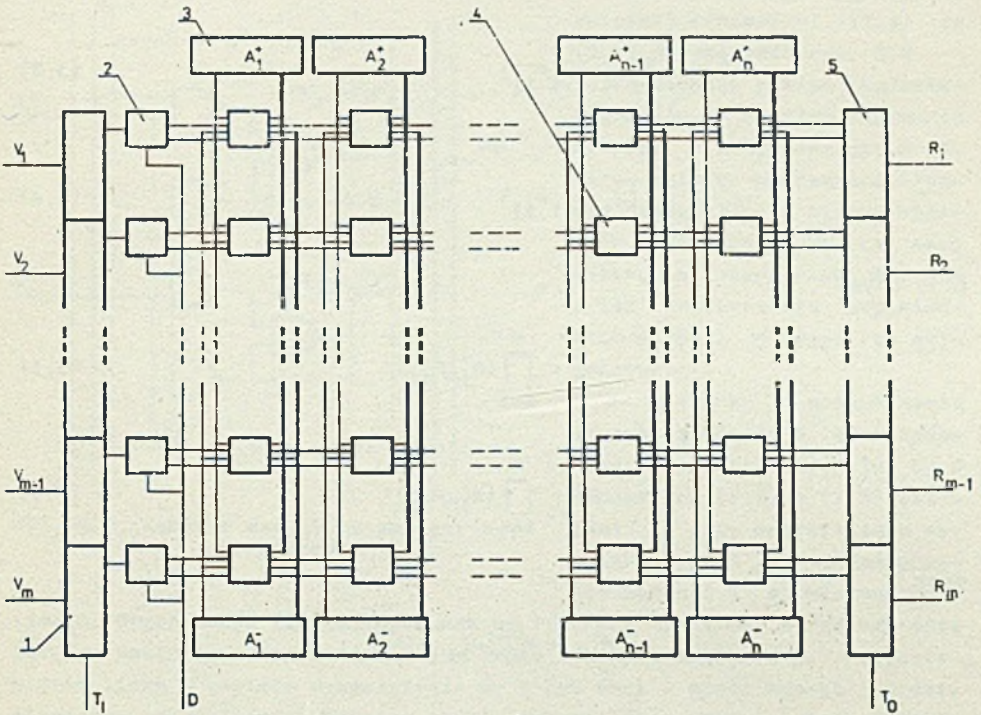
Algorytm poszukiwania liczby minimalnej jest dualny do przedstawionego. Przystosowanie go do tego celu wymaga zamiany w miejscach występowania stanu 1 stanem 0 i na odwrót.

Przedstawiony algorytm odnosił się do liczb dodatnich przedstawionych w skali rejestru. Jeżeli dopuścilibyśmy możliwość występowania w pamięci liczb ze znakiem, to pociągnęłoby to za sobą pewne skomplikowanie algorytmu. Korzystając wyłącznie z pamięci CAM o organizacji 4×2 (4 słowa 2-bitowe) lub 8×2 układ mikrosterujący należałoby wykonać z elementów dyskretnych co wydłużyłoby czas odpowiedzi całego układu do ułamków mikrosekundy.

Zaproponowano inną organizację pamięci skojarzeniowej.

3. PAMIĘĆ SKOJARZENIOWA WYSZUKUJĄCA W ZADANYM ZBIORZE ELEMENTY O WARTOŚCIACH EKSTREMALNYCH STEROWANA UKŁADOWO

W pamięci skojarzeniowej sterowanej układowo wykorzystano zasadę szeregowego porównywania między sobą tych samych bitów komórek pamięci tworzących zbiór, w którym ma być wskazany adres liczby ekstremalnej. Z każdym bitem pamięci skojarzeniowej związany został układ realizujący zadany algorytm pracy. Układ ten w dalszych rozważaniach będziemy nazywali komórką decyzyjną a ich zbiór w całej pamięci warstwą decyzyjną. Zbiór elementów pamiętających stan poszczególnych bitów słów zapisanych w pamięci będziemy nazywali warstwą pamiętającą. Do warstwy decyzyjnej zaliczymy również układy wybierające sygnał analizujący, wytwarzające sygnały decyzyjne oraz oba rejestry, wejściowy i wyjściowy. Rejestr wejściowy spełnia rolę rejestru adresowego o organizacji liniowej, co umożliwia zadanie operacji wyszukiwania w dowolnie wybranym zbiorze komórek pamięci.



Rys. 3.1. Schemat struktury połączeń między elementami pamięci skojarzeniowej

1 - rejestr wejściowy o wpisie asynchronicznym lub taktowanym sygnałem T_I , 2 - układ wybierający sygnał analizujący w powiązaniu z sygnałem kierunku poszukiwania D , 3 - układ generujący sygnał decyzyjny, 4 - komórka decyzyjna pamięci, 5 - rejestr wyjściowy synchronizowany sygnałem T_O

Ogólna struktura pamięci przedstawiona na rys. 3.1 nie zawiera warstwy pamiętającej wraz z układem zapisu i odczytu ze względu na to, że główny akcent przy opracowaniu pamięci skojarzeniowej został położony na warstwę decyzyjną. Od warstwy pamiętającej wymaga się tylko to, by miała charakter pamięci RAM i zapewniała bezpośredni dostęp warstwie decyzyjnej do każdego bitu pamiętanej informacji. Rola warstwy pamiętającej staje się istotna w niektórych zastosowaniach [4].

Podstawowym elementem omawianej pamięci skojarzeniowej jest jej komórka decyzyjna. Równania opisujące algorytm jej działania są następujące:

$$W_{ij}^+ = W_{i,j-1}^+ P_{ij} \vee W_{i,j-1}^+ A_j^+ \quad i = 1, 2, \dots, m \quad (3.1)$$

$$W_{ij}^- = W_{i,j-1}^- P_{ij}' \vee W_{i,j-1}^- A_j^- \quad j = 1, 2, \dots, n \quad (3.2)$$

$$A_j^+ = \prod_{i=1}^m (W_{i,j-1}^+ P_{ij}') \quad (3.3)$$

$$A_j^- = \prod_{i=1}^m (W_{i,j-1}^- P_{ij}') \quad j = 2, 3, \dots, n \quad (3.4)$$

przy czym:

$$A_1^+ = \prod_{i=1}^m (W_{i0}^+ P_{i1}') \quad (3.5)$$

$$A_1^- = \prod_{i=1}^m (W_{i0}^- P_{i1}') \quad (3.6)$$

oraz

$$W_{10}^+ = V_1^0 D (P_{11}' \vee A_1^-) \quad (3.7)$$

$$W_{10}^- = V_1^0 D' (P_{11}' \vee A_1^+) \quad i = 1, 2, \dots, m \quad (3.8)$$

gdzie:

W_{ij} - stan sygnału analizującego j-ty bit i-tej komórki pamięci skojarzeniowej,

P_{ij} - wartość j-tego bitu i-tej komórki pamięci, P_{i1} - oznacza bit znaku,

A_j - sygnał decyzyjny j-tej kolumny,

V_1^0 - sygnał z i-tej pozycji rejestru wejściowego,

D - sygnał kierunku szukania; $D = 1$ pamięć szuka wartości maksymalnej, $D = 0$ - pamięć szuka wartości minimalnej,

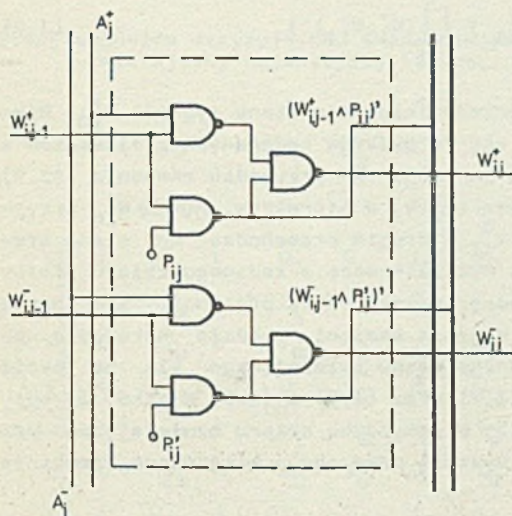
W_{10} - sygnał wyjściowy z układu wybierającego drogę dla sygnału analizującego $(\cdot)^+$ lub $(\cdot)^-$ zgodnie z poniższym oznaczeniem

$(\cdot)^+$ - oznaczenie sygnałów szukających wartości maksymalnej,

$(\cdot)^-$ - oznaczenie sygnałów szukających wartości minimalnej,

$(\cdot)'$ - negacja wyrażenia logicznego.

Uwaga - dla oznaczeń sygnałów przyjęto logikę pozytywną.



Rys. 3.2. Komórka decyzyjna pamięci skojarzeniowej

Budowę komórki decyzyjnej opisanej równaniami (3.1) do (3.4) przedstawia rys. 3.2.

Opracowana pamięć skojarzeniowa będzie działać poprawnie również, w przypadku gdy w zadanym zbiorze ekstremalna liczba znajdzie się w kilku komórkach. Pociągnie to za sobą wskazanie wszystkich adresów przez uaktywnienie odpowiednich pozycji w rejestrze wyjściowym.

O kierunku poszukiwania ekstremum decyduje stan zadanego z zewnątrz sygnału D . W równaniach (3.7) i (3.8) założono, że przy poszukiwaniu maksimum sygnał D powinien przyjąć wartość 1 a minimum wartość 0.

Organizacja pamięci pokazana na rys. 3.1 zakłada przedstawienie liczb w postaci stałoprzecinkowej w zapisie uzupełnieniu do 2. Przedstawienie liczb w zapisie uzupełnieniu do 1 lub znak - moduł wymaga korekty algorytmu poszukiwania. Wybrany stanem rejestru adresowego zbiór komórek pamięci skojarzeniowej może zawierać liczby wyłącznie dodatnie, wyłącznie ujemne lub liczby o różnych znakach. Opracowana pamięć poszukuje adresu liczby ekstremalnej zawsze w zbiorze liczb o tych samych znakach. Działając wg zależności ujętych równaniami (3.5) do (3.8) pamięć usuwa ze zbioru liczb o różnych znakach liczby ujemne przy poszukiwaniu adresu liczby największej i dodatnie przy określeniu adresu liczby najmniejszej. Wynika to z blokowania odpowiednich sygnałów W_{10} - równanie (3.7) lub (3.8) stanem sygnału A_1 - równanie (3.5) lub (3.6).

Poszukiwanie liczby ekstremalnej w zbiorze liczb ujemnych pamięć zamienia na poszukiwanie jej w zbiorze liczb dodatnich, traktując moduł liczby ujemnej łącznie ze znakiem jako moduł liczby dodatniej.

Działanie pamięci ilustrują przykłady podane w tablicy 1 i 2. W obu przypadkach została przyjęta pamięć o wymiarach 8×8 - osiem słów osmiorbitowych o organizacji bit znaku i siedem bitów modułu liczby. Rozpatrzymy przykład podany w tablicy 1.

W pamięci zostały umieszczone w podanej kolejności następujące liczby: 45, 109, 126, 8, -112, 87, -76, 121. Zadany został warunek poszukiwania maksimum oraz adres wybierający do zbioru 1, 2, 5, 7 i 8 komórkę pamięci. Działanie pamięci należy śledzić w oparciu o równania (3.1), (3.3), (3.6) i (3.7). Wstawmy (3.6) do (3.7), otrzymany

$$W_{10}^+ = V_1^0 D(P'_{11}) \vee \prod_{i=1}^m (W_{10}^+ P'_{i1}) \quad (3.9)$$

W równaniu (3.9) otrzymaliśmy przedefiniowanie stanu sygnału W_{10}^+ . Równanie to jak i poprzednie odnoszą się do układów budowanych z elementów o określonej szybkości działania, co powoduje w przypadku równania (3.9) rozbitcie stanu sygnałów W_{10}^+ na dwa takty. W pierwszym, sygnały W_{10}^+ przyjmują stany odpowiednich sygnałów V_1^0 . W drugim przechodzą do stanu stabilnego, w którym zostały wstępnie wyeliminowane z zadanego zbioru liczby ujemne. Rozbitcie sygnału analizującego na dwa takty przenosi się przez wszystkie komórki decyzyjne aż do wyjścia pamięci, gdzie wprowadza po pierwszym takcie zakłócenie poprawnego stanu ustalającego się po takcie drugim. Z analizy równań (3.6) i (3.7) oraz (3.5) i (3.8) wynika, że opisane rozbitcie na takty wystąpi tylko w przypadku zbioru zawierającego liczby o różnych znakach. Zjawisko to wystąpi przy obu kierunkach szukania ekstremum.

Tablica 1

Stany sygnałów decyzyjnych i analizujących szukających adresu komórki zawierającej największą liczbę w zadanym zbiorze - w tablicy podano również stany pośrednie tych sygnałów. * - oznacza adres szukanej liczby, ** - oznacza adres zakłócający stan wyjścia. Każdą komórkę decyzyjną opisuje para bitów $P_{ij} W_{ij}$

	$D=1$	A_1^+	A_2^+	A_3^+	A_4^+	A_5^+	A_6^+	A_7^+	A_8^+	
		1	1-0-1	1-0-0	1-0-0	1-1-0	1-1-1	1-1-1	1-1-0	
V_{10}^0	1	W_{10}^+	0-1-1	0-0-0	1-0-0	0-0-0	1-0-0	1-0-0	0-0-0	0
V_{10}^0	1	W_{10}^+	0-1-1	0-0-1	1-0-1	0-0-0	1-0-0	1-0-0	0-0-0	0
V_{10}^0	0	W_{10}^+	0-0-0	0-0-0	1-0-0	0-0-0	1-0-0	1-0-0	0-0-0	0
V_{10}^0	0	W_{10}^+	0-0-0	0-0-0	0-0-0	0-0-0	1-0-0	0-0-0	0-0-0	0
V_{10}^0	1	W_{10}^+	0-1-0	1-0-0	1-0-0	1-0-0	0-0-0	0-0-1	0-0-1	0
V_{10}^0	0	W_{10}^+	0-0-0	0-0-0	1-0-0	0-0-0	1-0-0	1-0-0	1-0-0	0
V_{10}^0	1	W_{10}^+	0-1-0	1-0-0	0-0-0	1-0-0	0-0-0	1-0-0	0-0-0	0
V_{10}^0	1	W_{10}^+	0-1-1	0-0-1	1-0-1	1-0-1	1-0-1	0-0-1	0-0-1	1
			1-0-0	1	1	1	1	1	1	
			A_1^-	A_2^-	A_3^-	A_4^-	A_5^-	A_6^-	A_7^-	A_8^-

W tablicy 2 podano przykład szukania adresu liczby najmniejszej. Wybrany przez adres zbiór zawiera liczby o różnych znakach. Nastąpi zatem podobne rozbitcie na takty jak w poprzednim przykładzie. Tablica 2 podaje tylko stan ustalony poszczególnych sygnałów. Na podstawie analizy równań oraz w oparciu o doświadczenie dostarczone przez poprzedni przykład można bez próby ustalenia stanów sygnałów w obu taktach podać, na której pozy-

Tablica 2

Stany sygnałów decyzyjnych i analizujących przy poszukiwaniu komórki zawierającej najmniejszą liczbę. * - wskazuje szukany adres

	D=0	A_1^+	A_2^+	A_3^+	A_4^+	A_5^+	A_6^+	A_7^+	A_8^+				
		0	1	1	1	1	1	1	1				
V_1^0	1	W_{10}^-	0	0 ₀	0 ₀	1 ₀	0 ₀	1 ₀	1 ₀	0 ₀	1 ₀	0	R_1
V_2^0	1	W_{20}^-	1	1 ₁	0 ₁	0 ₁	0 ₁	1 ₁	1 ₁	0 ₁	1 ₁	1	R_2^*
V_3^0	0	W_{30}^-	0	1 ₀	1 ₀	1 ₀	1 ₀	1 ₀	1 ₀	1 ₀	0 ₀	0	R_3
V_4^0	0	W_{40}^-	0	0 ₀	0 ₀	0 ₀	0 ₀	1 ₀	0 ₀	0 ₀	0 ₀	0	R_4
V_5^0	1	W_{50}^-	1	1 ₁	1 ₀	1 ₀	0 ₀	0 ₀	0 ₀	0 ₀	0 ₀	0	R_5
V_6^0	0	W_{60}^-	0	0 ₀	1 ₀	0 ₀	1 ₀	0 ₀	1 ₀	1 ₀	1 ₀	0	R_6
V_7^0	1	W_{70}^-	1	1 ₁	0 ₁	1 ₀	1 ₀	0 ₀	1 ₀	0 ₀	0 ₀	0	R_7
V_8^0	1	W_{80}^-	0	0 ₀	1 ₀	1 ₀	1 ₀	1 ₀	0 ₀	0 ₀	1 ₀	0	R_8
				1	0	0	0	1	1	0	1		
				A_1^-	A_2^-	A_3^-	A_4^-	A_5^-	A_6^-	A_7^-	A_8^-		

cji rejestru wyjściowego wystąpi nieprawidłowy stan wyjściowy po takcie pierwszym.

Określmy czas działania warstwy decyzyjnej. Interesujący przedział czasu będą wyznaczały dwa momenty: pierwszy, to przejście w stan aktywny sygnału W_{10} a drugi, osiągnięcie tego samego stanu przez sygnał W_{1n} . Sygnał analizujący przejdzie na swej drodze przez n komórek decyzyjnych. Oznaczmy (rys. 3.2) czas przejścia stanu aktywnego sygnału analizującego przez j -tą komórkę przy $P_{ij} = 0$ jako τ_g a przy $P_{ij} = 1$ jako τ_d . Oznaczenia dotyczą części komórki decyzyjnej poszukującej wartości maksymalnej. Założmy, że liczba ekstremalna zawiera zera na i bitach. Zauważmy ponadto, że eliminacja pewnych zakłóceń wynikających z pracy pamięci wg zadanego algorytmu wymaga, by $\tau_g > \tau_d$. Oznaczając τ_g/τ_d przez γ czas pracy warstwy decyzyjnej τ można określić z zależności

$$\tau = \tau_d [n + 1(\gamma - 1)] \quad (3.10)$$

Czas minimalny wynosi $n\tau_d/1=0$, czas maksymalny $n\tau_d/\gamma=1=n$. Iloraz tych dwóch czasów określa rozpiętość między najszybszą i najwolniejszą odpowiedzią pamięci. Wynosi ona γ . Ze względów przedstawionych powyżej powinna mieć miejsce zależność $\gamma \geq 2$. Najszybszą odpowiedź otrzymamy wtedy, gdy liczbą ekstremalną przy określaniu maksimum będzie -1 a najwol-

niejszą, gdy będzie nią 0. Zakładamy przy tym przedstawienie liczb w skali rejestru. Przy określaniu minimum liczby te zamienią się rolami.

4. PODSUMOWANIE

Opracowana wersja pamięci skojarzeniowej zapewnia, co stwierdzono na podstawie oszacowań, co najmniej o rząd większą szybkość działania niż wersja stosowana. Dla oszacowania szybkości przyjęto wykonanie obu pamięci wg tej samej technologii. Występujące zakłócenia, wynikające z rozbięcia w pewnych przypadkach stanu sygnału analizującego na dwa takty, można wyeliminować w różnych miejscach pamięci, tzn. zarówno w wyniku wytłumienia sygnału zakłócającego przed rozpoczęciem analizy pierwszej kolumny bitów, jak i też na drodze udostępniania stanu rejestru wyjściowego po czasie oszacowanym dla najgorszego przypadku.

Prezentowane rozwiązanie wymaga użycia w każdym układzie związanym z jednym bitem pamięci o jeden funktor więcej. Wynika to z przyjęcia w opracowanej wersji identycznego układu pamiętającego, zapisu i odczytu z użytym w wersji stosowanej. Porównanie odnosi się tylko do przypadku, gdy obie pamięci poszukiwałyby liczb ekstremalnych wyłącznie w zbiorze liczb dodatnich.

Dodatkową zaletą opracowanej pamięci jest możliwość dowolnego wybierania zbioru słów, wśród których ma być poszukiwany element ekstremalny. Można to realizować przez określanie stanu bitów w rejestrze wejściowym. Predystynuje to szczególnie podane rozwiązanie do zastosowań w roli arbitra [4].

W rozważaniach nad nową organizacją pamięci pominięto próby ostatecznego ustalenia pewnych rozwiązań. Do takich można zaliczyć te czy do określenia kierunku poszukiwania ekstremum używać jednej, czy dwóch linii. Jak łatwo się można przekonać obie wersje są w stanie zapewnić prawidłową pracę przy nieco różnym sterowaniu liniami interfejsu. Jeśli omawiane pamięci miałyby być wykonane w oparciu o układy scalone SSI i MSI, to pomijając fakt, że jest to opłacalne dla pamięci o małej pojemności, zależałoby nam na wersji jak najprostszej konstrukcyjnie. Rozważania te wiążą się z pojęciem technologiczności danego rozwiązania pod kątem wykonania go w postaci układu scalonego. W związku z tym ograniczono się do przedstawienia algorytmu pozostawiając opracowanie modułu pozwalającego na składowanie pamięci o zadanej pojemności jako temat wymagający odrębnych rozważań.

LITERATURA

- [1] T. Kohonen: Associative Memory, Communication and Cybernetics, Springer - Verlag, Berlin Heidelberg New York, vol. 17, 1977.
- [2] E. Nowak, Z. Sawicki: Pamięci maszyn cyfrowych, WNT, Warszawa 1977.
- [3] Pamięci półprzewodnikowe scalone, pod red. J. Eimbindera, WNT, Warszawa 1975.
- [4] M. Chmura: Zastosowanie pamięci skojarzeniowej w roli arbitra, Zeszyty Naukowe Politechniki Śląskiej seria Informatyka (w druku).

Recenzent

Doc. dr hab. Wiesław Traczyk

Wpłynęło do Redakcji 2.04.1980 r.

АССОЦИАТИВНОЕ ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО

Р е з ю м е

В статье представлены два решения ассоциативного запоминающего устройства. Первое — это общеприято в разных полупроводниковых системах образующих запоминающее устройство CAM. Второе представляет новую организацию очень быстродействующего ассоциативного запоминающего устройства. Оба решения представлены в отношении применения их с целью поиска экстремального элемента среди чисел сохраненных в памяти. Поиск осуществляется в нескольких шагах путем анализа последовательных битов двоичного числа, начиная с левого по одному сразу. Алгоритм поиска заключается в том, чтобы найти множество элементов, имеющих все данные выступить в роли экстремального элемента, среди которых в каждом шаге устраняется подмножество пока не останутся исключительно наименьшие или наибольшие.

ASSOCIATIVE MEMORY

S u m m a r y

The paper describes two different content-addressable memory organizations. The first is broadly used generally in various semiconductor circuit implementations of CAM, the second presents new organization of high speed associative memory. Both are described in application to searching for the extremal element among stored numbers. The search is carried out in several steps by studying the successive bits of the binary numbers, starting from the left, one bit at a time. The idea is to find a set of candidates for the extremum, from which each step eliminates a subset until only the largest or the smallest numbers are left.