Podstawy techniki cyfrowej zima 2020

Wykład 2

dr inż. Rafał Walkowiak

13.10.2020

Sposoby przedstawiania funkcji logicznych

- <u>Tablice Karnaugha</u>
- Kombinacji argumentów funkcji odpowiada pole tablicy, w polu umieszczamy właściwą dla kombinacji wartość.
- Sąsiednie (w poziomie i pionie – także cyklicznie) pola tablicy Karnaugha odpowiadają kombinacji argumentów różniącej się jedną wartością binarnej reprezentacji numeru kombinacji.
- Na rysunku zapisano kombinacje wejść – nie są to wartości funkcji.

		a	
b		0	1
	0	00	01
	1	10	11

		ba			
c		00	01	11	10
	0	000	001	011	010
	1	100	101	111	110

Tablica dla funkcji 2 i 3 zmiennych.

Sposoby przedstawiania funkcji logicznych

- Tablice Karnaugha – pola odpowiadające kombinacjom zmiennych

		a	
b		0	1
	0	00	01
	1	10	11

		ba			
c		00	01	11	10
	0	000	001	011	010
	1	100	101	111	110

		a	
b		0	1
	0	0	1
	1	2	3

		ba			
c		00	01	11	10
	0	0	1	3	2
	1	4	5	7	6

Wagi zmiennych użyte w numeracji kombinacji: a(1),b(2),c(3)

Reprezentacja funkcji logicznych za pomocą tablic Karnaugha

przykłady funkcji 2,3 i 4 zmiennych

b \ a	0	1
0	0	1
1	0	1

c \ ba	00	01	11	10
0	0	1	1	0
1	0	1	1	0

dc \ ba	00	01	11	10
00	1	1	1	1
01	0	∅	0	0
11	0	∅	0	0
10	1	1	0	1

∅ - oznaczenie wartości dowolnej funkcji

Sposoby przedstawiania funkcji logicznych

- Dysjunkcyjna (alternatywna) postać kanoniczna:

$$Y = f(x_0, x_1, \ldots, x_{n-1}) = \bigcup_{j=0}^{2^n - 1} a_j I_j$$

- Gdzie: U to suma
- I_j oznacza minterm - iloczyn zmiennych niezależnych dla j-tej kombinacji zmiennych =1; iloczyn zawiera zmienną prostą gdy w j-tej kombinacji bit zmiennej jest równy 1 lub zmienną zanegowaną gdy bit zmiennej jest równy 0
  - np. zerowa kombinacja wejść: 0000: minterm - $x_0' x_1' x_2' x_3'$ (wartość wyrażenia dla tej kombinacji wartości zmiennych wynosi jeden)
- a_j wartość funkcji odpowiadająca j-tej kombinacji zmiennych
- term – wyrażenie składające się ze zmiennych i symboli funkcyjnych
- **Por. Układy cyfrowe Wilkinson 3.1-3.4**

Dysjunkcyjna postać kanoniczna – przykład

Iloczyny zmiennych	abC_{in}	S	Cout
0 $a'b'c_{in}'$	000	0	0
1 $a'b'c_{in}$	001	1	0
2 $a'bc_{in}'$	010	1	0
3 $a'bc_{in}$	011	0	1
4 $ab'c_{in}'$	100	1	0
5 $ab'c_{in}$	101	0	1
6 abc_{in}'	110	0	1
7 abc_{in}	111	1	1

$S = 0a'b'c_{in}' + 1a'b'c_{in} + 1a'bc_{in}' + 0a'bc_{in} + 1ab'c_{in}' + 0ab'c_{in} + 0\ abc_{in}' + 1abc_{in}$

$S = a'b'c_{in} + a'bc_{in}' + ab'c_{in}' + abc_{in}$

$S = \cup(1,2,4,7)$ – gdzie liczby oznaczają numer kolejny kombinacji zmiennych dla której wartość funkcji =1

(należy określić wagę (1,2,4) zmiennej użytą do numeracji kombinacji; w przykładzie: **a** jest najbardziej znaczącym bitem - waga =4)

Sposoby przedstawiania funkcji logicznych

- Koniunkcyjna (iloczynowa) postać kanoniczna:

$$Y = f(x_0, x_1, \ldots, x_{n-1}) = \prod_{j=0}^{2^n-1} (a_j + S_j)$$

- Gdzie:
- S_j oznacza maxterm - sumę zmiennych niezależnych dla j-tej kombinacji zmiennych = 0; suma zawiera zmienną prostą gdy w j-tej kombinacji bit tej zmiennej jest równy 0 lub zmienną zanegowaną gdy bit kombinacji jest równy 1
  – Np. zerowa kombinacja wejść : 0000; suma dla tej kombinacji: $x_0+x_1+x_2+x_3$, wartość wyrażenia dla tej kombinacji zmiennych wynosi 0
- a_j oznacza wartość funkcji odpowiadającej j-tej kombinacji zmiennych.

Konjunkcyjna postać kanoniczna - przykład

sumy	abC_{in}	S	Cout
0 $a+b+c_{in}$	000	0	0
1 $a+b+c_{in}'$	001	1	0
2 $a+b'+c_{in}$	010	1	0
3 $a+b'+c_{in}'$	011	0	1
4 $a'+b+c_{in}$	100	1	0
5 $a'+b+c_{in}'$	101	0	1
6 $a'+b'+c_{in}$	110	0	1
7 $a'+b'+c_{in}'$	111	1	1

$S = (0+ a+b+c_{in}) (1+a+b+c_{in}')(1+a+b'+c_{in})(0+a+b'+c_{in}')$
$(1+a'+b+c_{in})(0+a'+b+c_{in}')(0+a'+b'+c_{in})(1+a'+b'+c_{in}')$

$S = (a+b+c_{in}) (a+b'+c_{in}')(a'+b+c_{in}')(a'+b'+c_{in})$

$S=\prod(0,3,5,6)$ – postać uproszczona iloczynu sum, gdzie liczby oznaczają numer kolejny kombinacji zmiennych, dla której wartość funkcji =0
(należy okreslić wagę (1,2,4) zmiennej użytą do numeracji kombinacji; w przykładzie: **a** jest najbardziej znaczącym bitem - waga =4)

Minimalizacja wyrażeń logicznych

- Postać kanoniczna nie jest najprostsza
- W optymalizacji – minimalizacji postaci wyrażeń logicznych używane kryteria kosztu to:
  - Redukcja liczby składników (lub czynników) wyrażenia funkcji (minimalizacja liczby bramek)
  - Redukcja liczby literałów(zmiennych) w wyrażeniu (minimalizacja liczby wejść bramek)
- Minimalizacja to przekształcanie postaci kanonicznej do postaci równoważnej – tańszej wg przyjętej funkcji kosztu
- Przykłady funkcji:
  - $f(a,b,c,d)= \cup(5,7,13,15)= d'cb'a+d'cba+dcb'a+dcba=ca$
  - Widoczna minimalizacja liczby składników wyrażenia z 4 do 1 i maksymalnej liczby literałów z 4 do 2
  - Zapis funkcji $f()= \cup(5,7,13,15)+d(1,3,4)$ oznacza brak konkretnego wymagania na wartość funkcji (dowolna wartość 0 lub 1) dla 1,3 i 4 kombinacji wejść.

Metoda minimalizacji siatki Karnaugha

- Założenia:
  - waga zmiennych ustalona np. : od najniższej wagi a,b,c,d
- Dla n zmiennych: Prostokątna tablica zawierająca 2^n pól, każde pole reprezentuje jeden minterm (maxterm), mintermy odpowiadające sąsiednim polom różnią się wartością tylko jednej zmiennej w reprezentacji binarnej kombinacji.

b \ a	0	1
0	0	1
1	2	3

c \ ba	00	01	11	10
0	0	1	3	2
1	4	5	7	6

dc \ ba	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

Twierdzenie o minimalizacji – reguła sklejania

- ab+ab’=a(b+b’)=a

b \ a	0	1
0	1	0
1	1	0

f(a,b)= Σ(0,2)= a’b’+a’b= a’(b+b’)=a’
Uzasadnienie do sklejania sąsiednich pól siatki. Powstającą grupę opisuje wyrażenie iloczynowe posiadające mniejszą liczbę zmiennych (usuwamy z opisu grupy tę zmienną, która dla pól przyjmuje różne wartości – b, a pozostaje a), zmienna, która dla grupy przyjmuje wartość 0 pozostaje w opisie (iloczynowym) grupy jako zmienna zanegowana (1 – zmienna prosta)

c \ ba	00	01	11	10
0	0	1	1	0
1	0	1	1	0

F(a,b,c)= Σ (1,3,5,7) = c’b’a+c’ba+cb’a+cba= c’a(b+b’)+ca(b+b’) = a (c+c’) =a
Sklejamy poziomo usuwając zmienną b, sklejamy pionowo usuwając zmienną c, pozostaje tylko **a** dla opisu grupy, a - proste (niezanegowane) gdyż przyjmuje wartość 1 dla wszystkich pól.

Twierdzenie o minimalizacji – reguła sklejania

- F(a,b) = Π(0,1)=(a’+b)(a+b)=a’a+a’b+ab+b=b

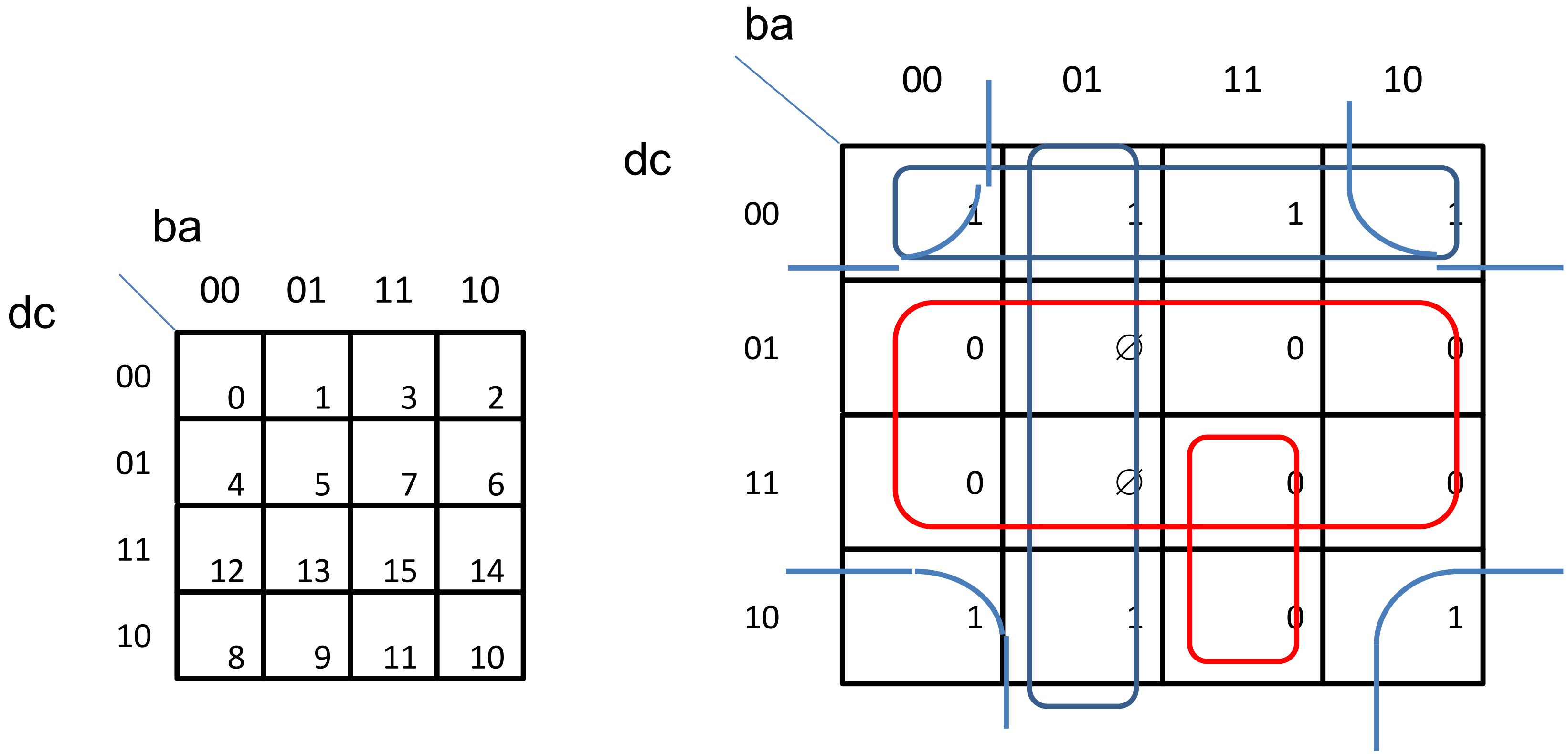
b \ a	0	1
0	0	0
1	1	1

Uzasadnienie do sklejania sąsiednich pól siatki. Powstającą grupę opisuje wyrażenie sumacyjne posiadające mniejszą liczbę zmiennych (usuwamy z opisu grupy tę zmienną, która dla sąsiednich pól przyjmuje różne wartości - a , a pozostaje b=0), zmienna, która dla grupy przyjmuje wartość 0 pozostaje w opisie grupy (sumacyjnym) jako zmienna prosta(gdy zmienna przyjmuje wartość 1 to w opisie grupy zmienna jest zanegowana).

c \ ba	00	01	11	10
0	0	1	1	0
1	0	1	1	0

F(a,b,c)= Π(0,2,4,6)=(a+b+c)(a+b’+c)(a+b+c’)(a+b’+c’) = a Sklejamy poziomo usuwając zmienną b, sklejamy pionowo usuwając zmienną c, dla opisu grupy pozostaje tylko a, proste gdyż przyjmuje wartość 0 (dla wszystkich pól).

Minimalizacja – sklejenia dla jedynek i zer



Funkcja f(a,b,c,d)= $\cup$(0,1,2,3,8,9,10)+d(5,13)

f=c’d’ +c’a’+b’a grupa pozioma, grupa narożna, grupa pionowa

f=c’ (d’+b’+a’)

Metoda tablic Karnaugha minimalizacji funkcji logicznej - formalnie

- TABLICE. Przygotowanie tablic dla danej liczby zmiennych i wpisanie wartości w polach. W polach w których wartość jest nieokreślona należy wpisać symbol nieokreslonoości np. ∅
- SKLEJENIA. Narysować obwiednie łączące pola tworzące możliwie największe obszary. Obwiednie łączą sąsiednie pola z jedynkami (dla postaci sumacyjnej funkcji) [pola z zerami (dla postaci iloczynowej funkcji)] . Sąsiedztwo także cykliczne. Obwiednie pokrywają grupy pól tworzące prostokąt (1,2,4,8,16… pól), każde **dwukrotne** powiększenie grupy wiąże się z usunięciem ze zbioru zmiennych charakteryzujących grupę jednej zmiennej (gdyż wartość tej zmiennej w nowej grupie jest dowolna).
- FUNKCJA. Zapisanie postaci minimalnej funkcji w oparciu o wykonane sklejenia (obwiednie), każde pole z 1 musi być pokryte przez dowolną grupę uwzględnioną w zapisie, grupa jest uwzględniona jako iloczyn zmiennych (postać sumy iloczynów) lub suma zmiennych (postać iloczynu sum) (wyrażenia te zapewniają żądaną wartość funkcji dla wszystkich kombinacji zmiennych z grupy).
- Uwaga: Pola ze znakami dowolnej wartości funkcji (∅) można łączyć z dowolnymi innymi polami (jedynek lub zer w zależności od postaci funkcji) dla uzyskania maksymalnych sklejeń.

Terminologia minimalizacji

- Implikant:
  - każdy minterm (iloczyn zmiennych) lub kombinacja zmiennych, dla których funkcja jest równa 1 lub
  - grupa mintermów dla których funkcja jest równa 1 lub Ø, które można skleić.
- Implikant prosty: implikant, którego nie można rozszerzyć przez sklejenia w tablicy Karnaugha.
- Implikant istotny: implikant prosty zawierający ten minterm (z warotścią funkcji =1), który nie występuje w żadnym innym implikancie prostym.

Terminologia minimalizacji

- Implicent:
  - każdy maxterm (suma zmiennych) lub kombinacja zmiennych, dla których funkcja jest równa 0 lub
  - grupa maxtermów dla których funkcja jest równa 0 lub Ø które można skleić.
- Implicent prosty: Implicent, którego nie można rozszerzyć przez sklejenia w tablicy Karnaugha.
- Implicent istotny: Implicent prosty zawierający ten maxterm (z wartością funkcji = 0), który nie występuje w żadnym innym implikancie prostym.

Metoda minimalizacji dwupoziomowej
(wersja suma iloczynów)

1. Wygeneruj wszystkie implikanty proste.
2. Utwórz pokrycie funkcji za pomocą minimalnej liczby implikantów (uwzględnij wartości funkcji =1).

Tablica pokrycia. Dla określenia postaci funkcji nie korzystającej wyłącznie z implikantów kluczowych należy wykonać tablicę pokrycia. Implikanty istotne (kluczowe) są koniecznymi elementami pokrycia funkcji.

Metoda minimalizacji dwupoziomowej
(wersja iloczynu sum)

1. Wygeneruj wszystkie implicenty proste.
2. Utwórz pokrycie funkcji za pomocą minimalnej liczby implicentów (uwzględnij wartości funkcji = 0).

Tablica pokrycia. Dla określenia postaci funkcji nie korzystającej wyłącznie z implicentów kluczowych należy wykonać tablicę pokrycia.

Implicenty istotne (kluczowe) są koniecznymi elementami pokrycia funkcji.

Przykład 1

dc \ ba	00	01	11	10
00	0	0	0	0
01	0	∅	1	0
11	1	∅	1	1
10	1	0	1	1

- Implikanty proste: ca, dc, db, da'
- Implikanty istotne: ca, da',db
- Implikanty istotne wystarczą do minimalnego pokrycia funkcji
- F(d,c,b,a)= ca+da'+db
- F(d,c,b,a)= (d+c)(a+d)(a'+b)

Przykład 1

- Realizacja funkcji na bramkach NAND bądź NOR
- przejście między rodzajami funkcji poprzez zastosowanie prawa deMorgana

$F(d,c,b,a)= (ca+da'+db)''=((ca)'(da')'(db)')'$

$F(d,c,b,a)= ((d+c)(a+d)(a'+b))''= ((d+c)'+(a+d)'+(a'+b)')'$