

154685

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**CMOS AKIM MODLU ÇOK SEVİYELİ MANTIK
DEVRELERİ TASARIMI**

YÜKSEK LİSANS TEZİ
Elk. Müh. Samir ÖZKAN
(504001397)

Tezin Enstitüye Verildiği Tarih : 26 Nisan 2004
Tezin Savunulduğu Tarih : 21 Mayıs 2004

Tez Danışmanı : Doç.Dr. Ali TOKER
Diğer Jüri Üyeleri Doç.Dr. Ece Olcay GÜNEŞ (İTÜ)
Doç.Dr. Coşkun SÖNMEZ (İTÜ)

Handwritten signatures of the thesis advisor and jury members.

MAYIS 2004

ÖNSÖZ

Tez çalışmam boyunca, vermiş olduğu destek için Elekt. Hab. Y. Müh. Ercan Altuntaş'a, mantıksal kavramlar konusundaki yardımlarından dolayı Yrd. Doç. Dr. Serhat İkizoğlu'na, her konuda sürekli destek olan değerli hocam Doç. Dr. Ali Toker'e teşekkürü borç bilirim...

Mayıs 2004

Samir ÖZKAN



İÇİNDEKİLER

TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
SEMBOL LİSTESİ	ix
ÖZET	xi
SUMMARY	xii
1. GİRİŞ	1
1.1. ÇSM Tarihçesi	3
1.2. ÇSM Uygulanabilen Teknolojiler	4
1.2.1. I ² L teknolojisi (Integrated Injection Logic)	4
1.2.2. ECL teknolojisi (Emitter Coupled Logic)	6
1.2.3. CCD teknolojisi (Charge Coupled Devices)	8
1.2.4. MOS teknolojisi	9
1.3. ÇSM Uygulama Alanları	10
1.3.1. Bellekler	10
1.3.1.1. ROM	10
1.3.1.2. Seri bellekler	10
1.3.1.3. RAM	10
1.3.2. Aritmetik uygulamalar	11
1.3.2.1. Çarpma devreleri	11
1.3.2.2. Toplayıcılar	12
1.3.3. Haberleşme arabirimleri	12
1.3.4. İşaret işleme	12
1.3.5. Diğer uygulamalar	12
2. ÇSM CEBRİ	14
2.1. ÇSM Operatörleri	15
2.1.1. Tek değişken (unary) operatörleri	15
2.1.1.1. Literal operatörü	16
2.1.1.2. Eşlenik literal operatörü	16
2.1.1.3. Saat yönü döngü operatörü (Clockwise Cycle Operator - CWC)	17
2.1.1.4. Ters saat yönü döngü operatörü (Counter Clockwise Cycle Operator)	17
2.1.2. İki ve çok değişkenli operatörler	18
2.1.2.1. Min operatörü	18
2.1.2.2. Sınırlı fark (Truncated Difference-Tdif) operatörü	18
2.1.3. Birleştirici operatörler	19
2.1.3.1. Max veya Sum operatörü	19
2.1.3.2. Modsum operatörü	20
2.1.3.3. Tsum (Truncated Sum) operatörü (sınırlı toplam)	20

2.2. ÇSM Fonksiyonlarının Cebirsel İfade Yöntemleri	22
2.2.1. ÇSM fonksiyonlarının polinom gösterimi	22
2.2.2. Reed-Müller kanonik formları	25
2.2.2.1. Max kanonik formu	25
2.2.2.2. Tsum kanonik formu	27
2.2.2.3. Genelleştirilmiş Reed-Müller kanonik formu	28
3. İNDİRGEME YÖNTEMLERİ	30
3.1. İndirgeme Üzerine Bazı Tanımlar	30
3.1.1. Minterm	31
3.1.2. Keyfi değer (don't care)	31
3.1.3. Mintermlerin kapsanması (cover)	31
3.1.4. Bileşen (implicant)	32
3.1.5. Asal bileşen (prime implicant)	32
3.2. Asal Bileşenlerle İndirgeme	32
3.2.1. Allen ve Givone indirgeme yöntemi	32
3.2.1.1. Konsensüs (birliktelik) işlemi (*)	33
3.2.2. Su ve Cheung indirgeme yöntemi	34
3.2.2.1. Kübik gösterilim	35
3.2.2.2. Asal bileşenlerle indirgeme yöntemi	38
3.2.2.3. İndirgenemez kapsam algoritması	38
3.3. Doğrudan Kapsama (Maskeleme) Yöntemleri	39
3.3.1. Pomper ve Armstrong algoritması	39
3.3.2. Besslich algoritması	41
3.3.2.1. "r" seviyeli endeksleme yöntemi	41
3.3.2.2. "r" seviyeli yer-değiştirme dönüşümü	42
3.3.2.3. "r" üzerinde değer atama dönüşümü (weight transform)	42
3.3.2.4. Kontrol tablosunun oluşturulması	44
3.3.2.5. Çarpan tarama dönüşümü	44
3.3.2.6. Bileşen seçimi	46
3.3.2.7. Max algoritması	48
3.3.2.8. Plus (Tsum) algoritması	48
3.3.3. Dueck&Miller indirgeme yöntemi	49
3.3.4. İndirgeme yöntemlerinin karşılaştırılması	49
3.3.5. Maliyet tabloları ile indirgeme teknikleri	50
4. ÇSM CMOS AKIM MODLU DEVRE TASARIMI	51
4.1. Temel Devreler	51
4.1.1. Akımların lineer toplanması (Sum)	53
4.1.2. Sabit akım kaynağı (Constant)	53
4.1.3. NMOS ve PMOS akım aynaları	53
4.1.4. Gerilim çıkışlı akım karşılaştırıcı (Thershold) devresi	55
4.1.5. Basit anahtar (Switch)	56
4.2. ÇSM Operatörlerinin Devre Gerçeklemeleri	56
4.2.1. Hibrit CMOS ÇSM devreleri	56
4.2.1.1. Min operatörünün devre gerçeklemesi	58
4.2.1.2. Tsum operatörünün (sınırlı toplam) devre gerçeklemesi	60

4.2.1.3.	Literal operatörünün devre gerçektelemesi	61
4.2.1.4.	Eşlenik Literal operatörü gerçektelemesi	63
4.2.1.5.	Döngü operatörü	65
4.2.1.6.	ÇSM akım yenileyici	67
4.2.1.7.	İki seviye-dört seviye dönüştürücü (kodlayıcı)	69
4.2.1.8.	Dört-iki seviyeli dönüştürücü (kod çözücü)	71
4.2.2.	Tam akım modlu devreler	74
4.2.2.1.	Tdif operatörünün devre gerçektelemesi	74
4.2.2.2.	Akım çıkışlı akım karşılaştırıcılar	75
4.2.2.3.	ÇSM akım evirici devresinin tasarımı	76
4.2.2.4.	Max operatörünün devre gerçektelemesi	77
4.2.2.5.	Min operatörünün devre gerçektelemesi	79
4.2.2.6.	Saat yönü Döngü operatörünün devre gerçektelemesi (CWC)	80
4.2.2.7.	Literal operatörü'nün devre gerçektelemesi	81
4.2.2.8.	Tsum operatörünün devre gerçektelemesi	82
5.	ÖRNEK FONKSİYON TASARIMI	84
6.	SONUÇLAR VE DEĞERLENDİRME	91
	KAYNAKLAR	92
	EK A	97
	ÖZGEÇMİŞ	99

TABLO LİSTESİ

	<u>Sayfa no</u>
Tablo 1.1 Çarpma devreleri karşılaştırması	11
Tablo 2.1 Sayı tabanına göre fonksiyon sayısı	15
Tablo 2.2 Bazı Literallerin doğruluk tablosu	17
Tablo 2.3 Min operatörünün doğruluk tablosu	18
Tablo 2.4 Sınırlı fark operatörünün doğruluk tablosu	19
Tablo 2.5 Max operatörünün doğruluk tablosu	19
Tablo 2.6 Modsum operatörünün doğruluk tablosu	20
Tablo 2.7 Sınırlı toplam operatörünün doğruluk tablosu	20
Tablo 2.8 Yaygın olarak kullanılan ÇSM operatörleri	21
Tablo 2.9 İki değişkenli dört seviyeli fonksiyon	26
Tablo 2.10 İki değişkenli örnek fonksiyon	28
Tablo 3.1 İki değişkenli dört değerli fonksiyonun doğruluk tablosu	31
Tablo 3.2 İki değişkenli fonksiyonun tablo gösterilimi	33
Tablo 3.3 Üç değişkenli endeksleme tablosu	41
Tablo 3.4 Minterm ve bileşen seçim tablosu	50
Tablo 4.1 Temel devreler	52
Tablo 4.2 Literal operatörleri için transistör boyutları	64
Tablo 4.3 Hibrit Döngü operatörü için transistör boyutları	66
Tablo 4.4 İkili-Dört seviye dönüşümü doğruluk tablosu	70
Tablo 5.1 Örnek fonksiyon doğruluk tablosu	84
Tablo 5.2 Tasarım algoritmalarının karşılaştırılması	90

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1	Temel I^2L kapısı 4
Şekil 1.2	I^2L akım aynası 4
Şekil 1.3	I^2L lineer akım toplayıcı 5
Şekil 1.4	I^2L Seviye yenileyici 5
Şekil 1.5	Temel ECL akım anahtarı 7
Şekil 1.6	ECL ÇSM akım evirici 7
Şekil 1.7	CCD elemanı 8
Şekil 1.8	CCD yük kuyusu 9
Şekil 3.1	İşaret akış diyagramı ve yerdeğiştirme örneği 43
Şekil 3.2	Değer atama dönüşümü 45
Şekil 3.3	00 endeksinin bileşen ve asal bileşenleri 46
Şekil 3.4	Üç değerli dönüşüm işaret akış diyagramı 47
Şekil 4.1	Basit CMOS akım aynası 54
Şekil 4.2	Hibrit devre tasarım yaklaşımı blok diyagramı 57
Şekil 4.3	Giriş (a) ve çıkış (b) devre örnekleri 58
Şekil 4.4	Hibrit Min operatörü devresi 59
Şekil 4.5	Hibrit Min operatörü simülasyon sonuçları 59
Şekil 4.6	Hibrit Tsum operatörü devresi 60
Şekil 4.7	Hibrit Tsum operatörü simülasyon sonuçları 61
Şekil 4.8	Hibrit Literal operatörü devresi 61
Şekil 4.9	Hibrit Literal operatörü simülasyon sonuçları 62
Şekil 4.10	Hibrit eşlenik Literal operatörü devresi 63
Şekil 4.11	Hibrit eşlenik Literal operatörü simülasyon sonuçları 64
Şekil 4.12	Hibrit Döngü operatörü devresi 65
Şekil 4.13	Hibrit Döngü operatörü simülasyon sonuçları 65
Şekil 4.14	Hibrit ÇSM akım yenileyici 67
Şekil 4.15	Geliştirilmiş Hibrit ÇSM akım yenileyici devresi 69
Şekil 4.16	Geliştirilmiş Hibrit ÇSM akım yenileyici simülasyon sonuçları 69
Şekil 4.17	İkili-Dört seviyeli kodlayıcı 70
Şekil 4.18	İkili-Dört seviyeli kodlayıcı simülasyon sonuçları 71
Şekil 4.19	Dört-iki seviyeli kod çözücü devresi 72
Şekil 4.20	Dört seviyeli ikili kod çözücü simülasyon sonuçları 72
Şekil 4.21	Kod çözücü ikili kapı bloğu devresi 73
Şekil 4.22	Tam akım modlu Tdif operatörü devresi ve sembolü 74
Şekil 4.23	Tam akım modlu Tdif operatörü simülasyon sonuçları 74
Şekil 4.24	Yüksek akım karşılaştırmacı devresi ve sembolü 75
Şekil 4.25	Yüksek akım karşılaştırmacı simülasyon sonuçları 76
Şekil 4.26	Akım evirici devresi (a) ve sembolü (b) 77
Şekil 4.27	Akım evirici simülasyon sonuçları 77
Şekil 4.28	Max operatörü devresi, sembolü ve blok diyagramı 78
Şekil 4.29	Max operatörü simülasyon sonuçları 78

Şekil 4.30	Min operatörü devresi, sembolü ve blok diyagramı	79
Şekil 4.31	Min operatörü simülasyon sonuçları	79
Şekil 4.32	Döngü operatörü devresi ve sembolü	80
Şekil 4.33	Döngü operatörü simülasyon sonuçları	80
Şekil 4.34	Literal operatörü blok diyagramı ve sembolü	81
Şekil 4.35	Literal operatörü simülasyon sonuçları	82
Şekil 4.36	Tam akım modlu Tsum operatörü simülasyon sonuçları	82
Şekil 5.1	Hibrit Tsum algoritması ile örnek fonksiyonun blok diyagramı	87
Şekil 5.2	Max algoritması ile örnek fonksiyonun blok diyagramı	88
Şekil 5.3	Örnek fonksiyonun Max gerçekleştirmesinin simülasyonu	88
Şekil 5.4	Tsum algoritması ile örnek fonksiyonun blok diyagramı.....	89
Şekil 5.5	Örnek fonksiyonun Tsum gerçekleştirmesinin simülasyonu.....	90



SEMBOL LİSTESİ

r	:ÇSM sayı tabanı (radix)
b	:Basamak sayısı (digit)
C	:Karmaşıklık maliyeti
Φ_s	:Kuyu potansiyeli
Q_s	:Kuyu yükü
Q_E	:Birim yük
t	:CCD Transfer kapısı
s	:CCD Depolama kapısı
$\bullet$	:Min operatörü
$\cup, +$	:Max veya Sum operatörü
$\boxplus$	:Sınırlı toplam operatörü (Tsum)
$\boxminus$	:Sınırlı fark operatörü (Tdif)
$\sum$	:Mantık toplama işlemi
$\prod$	:Mantık çarpma işlemi
$k \left[\begin{smallmatrix} a \\ \{x\}^b \end{smallmatrix} \right]$	:k değerli Literal operatörü
$k \left[\overline{\begin{smallmatrix} a \\ \{x\}^b \end{smallmatrix}} \right]$	:k değerli eşlenik Literal
$x \xrightarrow{a}$	:Saat yönü Döngü operatörü
d	:Keyfi değer (don't care)
$P(x_1, x_2, \dots, x_n)$	:Çarpan (product)
$I(x_1, x_2, \dots, x_n)$	:Bileşen
$I'(x_1, x_2, \dots, x_n)$	:Asal Bileşen
$*$	:Konsensüs (birliktelik) işlemi
A^*	:Katsayısız kübik terim
$\underline{A^*}$	: A^* elemanlarından oluşan kübik vektör
U	:Kübik alan birleşimi
$\#$	:Kübik sharp işlemi
$\cap$	:Kübik alanların kesişimi
M	:ÇSM fonksiyonlarını kübik notasyona dönüştürme işlemi
R	:M dönüşümünün tersi
$\emptyset$	:Boş küme
$\underline{PI_i}$	:i elemanlı asl bileşenler kümesi
T	:Kübik geçici kapsam
$\bar{i}_m$	:Yerdeğiştirme endeksi
D	:İki endeks arasındaki genelleştirilmiş Hamming mesafesi
$\Delta(a, b)$	:a ve b değişkenleri arasındaki endeks mesafesi
$w(x)$	:Değer atama işlemi

B	:Değer atama çarpanı
e_i	:Etkinlik katsayısı
S	:Endeks yer-değiştirme işlemi
m_i	:Kapsama derecesi
W	:MOS transistor kanal genişliği
L	:MOS transistor kanal uzunluğu
λ	:Kanal boyu modülasyonu parametresi
V_{GS}	:MOS transistorda kapı-kaynak gerilimi
V_t	:MOS transistorda eşik gerilimi
g_m	:MOS transistorun geçiş iletkenliği
C_{gs}	:MOS transistorda kapı-kaynak parazitik kapasitesi
C_{ox}	:MOS transistorda oksit parazitik kapasitesi
μ	:Yüzey hareket kabiliyeti
τ	:Zaman sabiti
t_p	:Propagasyon gecikmesi
I, I_b, I_0	:Taban akımı



ÖZET

Çok Seviyeli Mantık (ÇSM), ikili mantık sisteminin genelleştirilmiş halidir. ÇSM ile birçok işlemin daha az transistör ile yapılabileceği kabul edilmektedir. Ancak ÇSM ile fonksiyon gerçekleştirme kayda değer derecede karmaşıktır.

Bu çalışmada, ÇSM fonksiyonlarını daha az transistör ile gerçeklemeye elverişli yeni devrelerin tasarımı amaçlanmıştır. Bu amaca yönelik olarak temel indirgeme yöntemleri ve mevcut iki CMOS devre tasarım tekniği incelenmiştir. CMOS teknolojisi ile ilk ÇSM devre tasarım örnekleri Hibrit olarak nitelenen devre topolojisi üzerine kuruludur. Bu tasarım yönteminde, akım modlu giriş ve çıkış devreleri ile gerilim modlu kontrol katı aynı yapı bloğu içerisindedir. Ancak bu tasarım tekniği, gerilim modlu kontrol katındaki gecikmelerden dolayı işaret bozulması fazla olduğundan, günümüzde pek tercih edilmemektedir. Ayrıca bu tasarım yaklaşımı yüksek sayı tabanı kullanımına uygun düşmemektedir. Zira, sayı tabanı ile devre karmaşıklığı hızla artmaktadır. Modern tasarım yaklaşımı, tamamen akım modlu çalışan devreler üzerine kuruludur. Bu yapıda, aynı operatörü tasarlamak için gereken transistör sayısı hibrit modlu tasarlanan devreden yaklaşık % 50 daha azdır.

Sınırlı toplam operatörü (Tsum) ile fonksiyon gerçekleştirilmesi durumunda, indirgeme olasılığının Max operatörü ile tasarlanmış fonksiyona göre daha fazla olduğu bilinmektedir. Buradan hareketle, daha önce Hibrit modlu olarak tasarlanan Tsum operatörü, tam akım modlu Min operatörü devresi revize edilerek yeniden tasarlanmıştır. Hibrit modlu olarak 9 transistör ile oluşturulabilen bu operatör, tam akım modlu olarak 4 transistör ile gerçekleştirilmiştir. Tsum ile indirgeme olasılığının yüksek oluşu ve operatör tasarlamak için gereken transistör sayısının az oluşu, bir fonksiyonu gerçeklemek için gereken transistör sayısını azaltmıştır. Bu durum bir örnek fonksiyon üzerinde gösterilmiş ve simülasyon sonuçları da verilmiştir.

SUMMARY

Multiple Valued Logic (MVL), is generalized form of binary logic system. It is accepted that with MVL many operations can be made with fewer transistors. But function realisation with MVL is quite complex.

In this study, new circuit's designs are aimed which are suitable for realizing MVL functions with fewer transistors. Inclined towards this aim, basic minimization methods and existing two CMOS circuit design techniques examined. First MVL circuits designs, made with CMOS technology are based on Hybrid circuit topology. In this design topology, current mode input and output blocks and voltage mode control block are placed in the same circuit. But this design method isn't preferred today due to signal deteriorations, because of voltage mode control block delays. In addition this design approach doesn't comply with using higher radix systems. Because with radix (number system) increase complexity increases rapidly. Modern design approach is based on full current mode circuits. In this method, it is needed nearly %30 fewer transistors than hybrid mode to design the same operator.

In case of function realisation with Truncated Sum operator (Tsum), it is known that the minimization possibility is much more than function realisation with Max operator. Starting from this point, in hybrid mode beforehand designed Tsum as hybrid mode is redesigned by revising the full current mode Min operator circuit. While this operator can be formed with 9 transistors as hybrid mode, it is realised with 4 transistors as full current mode. Because of high minimisation possibility and less transistor requirements for operator realisations, transistor number for realising a full function decreases. This situation is shown on a sample function and simulation results are included.

1. GİRİŞ

İkili sayı sistemleri ile çalışan mantık devreleri günümüzde oldukça karmaşık bir yapıya ulaşmıştır. Fonksiyonların cebirsel ifadesi ve indirgeme yöntemleri basit ve güçlü bir yapıya sahiptir. Ayrıca tasarlanan devreler de oldukça kararlı çalışmaktadır. Ancak fonksiyon karmaşıklığı arttıkça, devrelerin fiziksel olarak gerçekleşme zorluğu da o oranda artmıştır. Özellikle gerçekleşen yongalar arasında ve yonga ile dış dünya (dış kenar) arasındaki bağlantılar tasarımı zorlaştıran en önemli etkenlerdir. Tasarlanan karmaşık devrelerin boyutları o denli artmıştır ki IBM tarafından 200 pinli özel kılıf tasarlanmıştır [1]. Düğümler için kullanılan malzeme alanı birçok durumda aktif devre için kullanılan malzeme alanını aşmaktadır. Deneyimler göstermiştir ki kenar bağlantıları için kullanılan malzeme alanı n iken, devre için kullanılan malzeme alanı n^2 olmaktadır. Ancak devre üzerinde durum çok farklıdır, iş gören aktif devrenin alanı n^2 iken, iç düğümler için kullanılan malzeme alanı yaklaşık $(n^2)!$ ile verilmektedir. Dolayısıyla tasarım için harcanan çaba ve maliyet, devre karmaşıklığı ile artmaktadır [2]. Elektronik devre teknolojilerinde fiziksel veya mali kısıtlamalar, devrelerin küçültülmesine engel oluşturmaktadır. Bu açıdan bakıldığında devre boyutlarını küçültmek için fiziksel küçültmenin yanısıra fonksiyon esnekliği de düşünülerek devre ebatları küçültülebilir [3].

Entegre devre teknolojisindeki gelişmeler ikiden fazla mantık seviyesini kullanan devrelerin tasarımına olanak sağlamıştır. İkiden fazla ayrık işaret işleyebilen mantık devrelerine “Çok Seviyeli Mantık” (ÇSM) (Multiple Valued Logic MVL) adı verilmektedir [4]. Bu konudaki öncü çalışma 1921 yılında Post tarafından yapılmıştır. Post cebri herhangi bir sayı tabanına uyarlanabilmektedir. İlk devre tasarım örneklerine 1950lerde rastlanmaktadır. Ancak tasarlanan devrelerin karasız oluşu ve sağlam bir matematik modele dayanmayışı konu üzerinde ilgiyi azaltmıştır [5]. Post cebri üzerine bugüne kadar pekçok araştırma yapılmıştır. Ancak bunların birçoğu devre tasarımına uygun düşmemektedir. Devre tasarımına uygun düşen ilk cebri ifade yöntemi 1956 yılında Lee ve Chen tarafından verilmiştir [6]. Özellikle

Allen-Givone cebri, fiziksel olarak gerçekleştirilecek operatörlerden oluşması, indirgemeye uygun olması bakımından ÇSM alanında en önemli gelişme olarak kabul edilmektedir [7].

Herhangi bir nümerik sistemde, verilen bir büyüklüğü ifade etmek için kullanılan sayı tabanı arttıkça gereken basamak sayısı azalmaktadır. Bir N sayısını ifade etmek için gereken basamak sayısı $N=r^D$ ile verilmektedir. Burada r sayı tabanı (radix) D de gereken basamak sayısıdır. Bunun ötesinde karmaşıklık maliyeti C basit olarak şöyle ifade edilebilir;

$$C=k(r.D)=k \left[r \frac{\log N}{\log r} \right] \quad (1.1)$$

Burada k herhangi bir sabiti göstermektedir. Bir işareti işleme maliyetinin sayı tabanı dikkate alınmaksızın sabit kaldığı düşünülürse toplam sistem maliyetinin D ile orantılı olduğu görülür. Sayı tabanı arttıkça maliyet azalmaktadır [5].

Ancak ÇSM'nin bu avantajlarına rağmen, fonksiyon tasarımı hala karmaşıklığını korumaktadır. İndirgeme yöntemleri de henüz tatmin edici sonuçlar vermemektedir. Fakat 1980'lerden sonra konu üzerindeki araştırmaların artması ve birçok devrenin fiziksel olarak gerçekleştirilebilir hale gelmesi konu üzerindeki ilgiyi arttırmıştır [5, 8].

ÇSM ile bugüne kadar çok sayıda operatör (fonksiyon) tanımlanmıştır. Bunların bazıları ikili operatörlerin (kapıların) karşılığıdır. Ancak ÇSM'de fonksiyonların gerçekleştirilebilirliği teknoloji bağımlıdır. Dolayısıyla her operatör ÇSM'ye uygulanabilen her teknoloji ile gerçekleştirilememektedir. Bu durumun sonucu olarak da indirgeme yöntemleri de teknoloji bağımlı olmaktadır [2].

Bu çalışmada, ÇSM matematik modelleri ve mevcut devre yapılarını incelemek ve mevcut devreleri geliştirme amaçlanmıştır.

Giriş bölümünde ÇSM tanıtılmış ve tarihi gelişimi ile son dönemde yoğun ilgi odağı olmasının sebepleri irdelenmiştir. Ayrıca ÇSM'ye uygulanabilen transistor teknolojileri ana hatları ile verilmiş ve uygulama örnekleri özetlenmiştir.

İkinci Bölümde, operatörlerin ve bütün fonksiyonun matematik modelleri ayrıntılı olarak verilmiştir.

Üçüncü Bölümde, en çok kullanılan indirgeme yöntemleri incelenmiş ve indirgeme terminolojisi örneklerle verilmiştir.

Dördüncü Bölümde, hibrit ve tam akım modlu devre tasarımı incelenmiştir. Ayrıca mevcut devreler CMOS UMC018 teknolojisi ile yeniden oluşturulmuş ve çalışma prensipleri ayrıntılı olarak verilmiştir. Yeni bir tam akım modlu Tsum operatörü Min operatöründen türetilmiştir.

Beşinci Bölümde, değişik algoritmalar ile bir örnek fonksiyon gerçekleştirilmiştir.

1.1. ÇSM Tarihçesi

Daha önce de bahsedildiği üzere ÇSM'ye uyarlanabilen matematik model 1921'de Post tarafından önerilmiştir. Daha sonra Post cebrinin üç seviyeli mantığa uygun düşen cebri ifadeleri Lee ve Chen tarafından önerilmiştir [6]. ÇSM cebrinde en esnek operatör olan Literal, Allen ve Givone tarafından 1968'de önerilmiştir [7]. Allen ve Givone tarafından önerilen cebri ifade yöntemi en yaygın kullanılan ÇSM cebridir. İzleyen yıllardaki araştırmalar ise ağırlıklı olarak ÇSM fonksiyonlarının indirgenmesi üzerine yapılmıştır. İndirgeme alanındaki önemli çalışmalar 1979 Miller ve Muzio [9], Pomper ve Armstrong 1981 [10], Beslich 1986 [11], Dueck ve Miller 1987 [12] tarafından yapılan çalışmalardır.

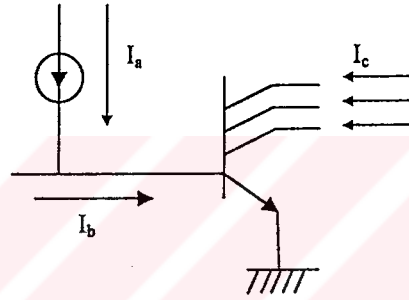
Devre tasarım örneklerine 1950'lerde rastlamak mümkündür. Üç seviyeli ÇSM mantığı kullanan Setun adlı bilgisayar 1956 yılında Rus bilim adamları tarafından hayata geçirilmiştir. 70'li yıllara kadar bipolar teknolojisi ile gerçekleştirilmiş devreler ağırlıktadır. O tarihten günümüze kadar ki dönemde ise CCD (Charge Coupled Devices) ve MOS teknolojisi ile gerçekleştirilmiş devreler ön plana çıkmıştır [5]. VLSI teknolojisindeki gelişmelere paralel olarak MOS teknolojisi ile gerçekleştirilen devrelere ilgi giderek artmaktadır.

1.2. ÇSM Uygulanabilen Teknolojiler

ÇSM'ye, ikili'ye uygulanabilen hemen-hemen bütün teknolojiler uygulanabilmektedir. Zaten ÇSM devrelerinin birçoğu ikili devrelerin ÇSM'a uyarlanması ile elde edilmiştir. ÇSM'ye uygulanabilen teknolojiler genel hatları ile aşağıda verilmiştir.

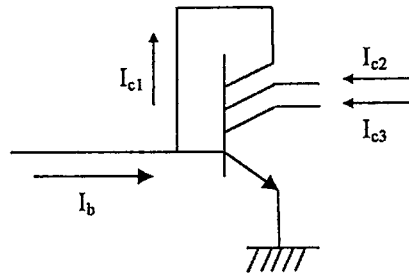
1.2.1. I^2L teknolojisi (Integrated Injection Logic)

Şekil 1.1'de I^2L teknolojisinin temel yapı bloğu verilmiştir [2]. Devre akım çıkışlı akım karşılaştırıcıdır.



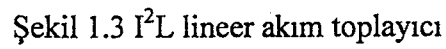
Şekil 1.1 Temel I^2L kapısı

Şekil 1.2'de ise çok kollektörlü I^2L akım aynası verilmiştir [2];



Şekil 1.2 I^2L akım aynası

Şekil 1.3'te I^2L lineer akım toplayıcı devresi verilmiştir [2]. Bu devre, çok kollektörlü transistörlerin kollektörlerinin birkaçının birleştirilmesi ile giriş akımının birleştirilen kollektör sayısına çarpımını almakta ve diğer transistör girişi ile lineer toplamını almaktadır.



5

$2.5I_0$ akım seviyeleri için karşılaştırma görevi görmektedir. T5, T6 ve T7 transistörleri da, akım çıkışlı akım karşılaştırıcıların çıkış işaretinin eşleniğini alarak akım seviyelerini yeniden oluşturmaktadır. Bu transistörlerin kollektörleri birleştirilerek x giriş akımı seviyeleri yeniden elde edilmektedir. T8 transistörü ise sadece akım yönünü değiştirmek için kullanılmıştır.

I^2L teknolojisi ile bazı işlemler kolayca gerçekleştirilebilmektedir. Bunlar aşağıdaki gibi sıralanabilir;

- Akım aynalanması
- Akımların lineer toplanması
- Gerilim çıkışlı akım karşılaştırma (thresholding)

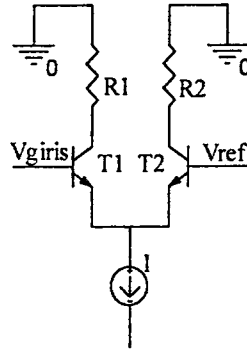
Akım aynalanması ve toplayıcı devreleri yukarıdaki şekillerde verilmiştir. I^2L ile akım aynalaması, akımların toplanması, akımların işaret değiştirilmesi vb. kolay olduğundan genellikle akım modlu çalışan devrelerin tasarımında kullanılmışlardır. Ancak literatürde pasif elemanlarla birlikte tasarlanmış gerilim modlu devrelere de rastlamak mümkündür [2].

1.2.2. ECL teknolojisi (Emitter Coupled Logic)

Yüksek hızlarda çalışabilmeleri dolayısı ile I^2L ve ECL yoğun araştırma konusu olmaktadır. Bipolar teknolojisinin bu iki üyesi için kısa bir değerlendirme yapmak gerekirse; ECL tekniği ile, I^2L teknolojisi ile gerçekleştirilebilen bazı operatörler gerçekleştirilebilmektedir ancak devreler biraz daha karmaşık olmaktadır. Ancak karmaşık olmasının yanı sıra hızları I^2L operatörlerine göre yüksektir [13,14]. Diğer üstün yanları da hem akım modlu hem de gerilim modlu olarak kullanılabilir olmalarıdır.

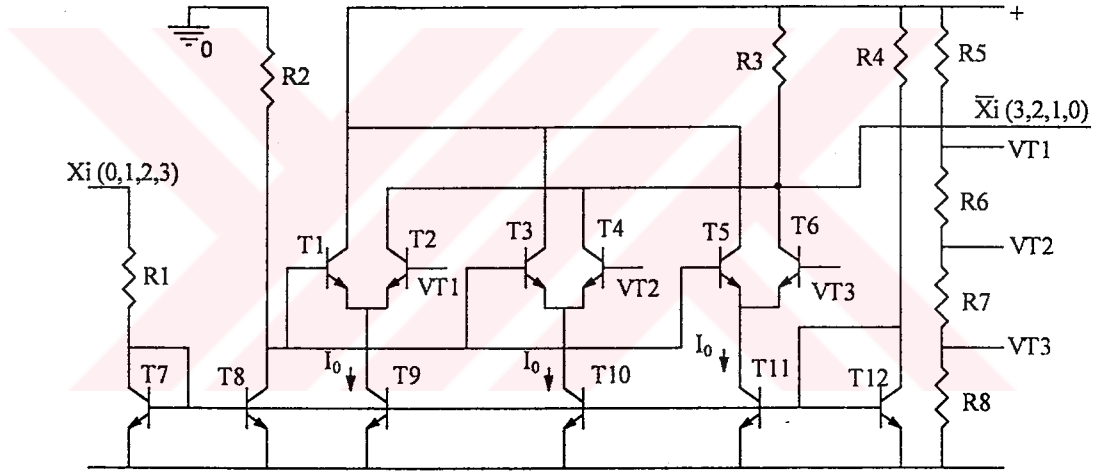
Şekil 1.5'te temel ECL devresi verilmiştir. Devrenin giriş gerilimi V_{ref} 'ten küçük ise I akımını R2 üzerinden akar, eğer büyükse R1 üzerinden akar.

Bu devre temel olarak iki seviyeli mantık içindir ve gerilim karşılaştırıcı olarak görev yapmaktadır. ÇSM için ise, birkaç akım birleştirilir ve direnç ile gerilime çevrilir veya tam tersi işlemle direnç ile gerilim akıma çevrilir [2].



Şekil 1.5 Temel ECL akım anahtarı

ECL teknolojisi ile ÇSM devre tasarımına ilişkin bir örnek devre Şekil 1.6'da verilmiştir [2]. Devre akım modlu eviricidir. Çalışma prensibi dördüncü bölümde verilecek olan CMOS devrelerin çalışmasına benzemektedir.



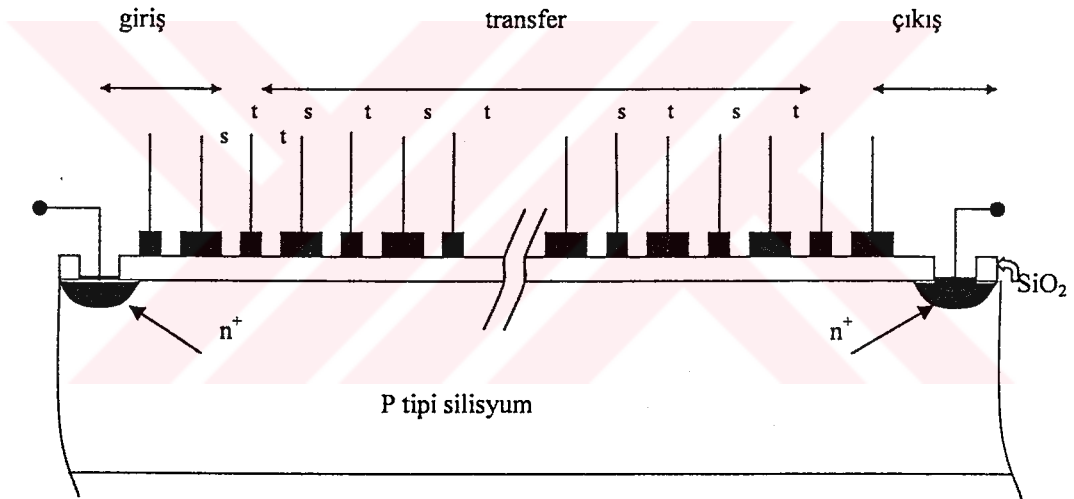
Şekil 1.6 ECL ÇSM akım evirici

Şekil 1.6'daki devrede T7 ve T8 transistörleri akım aynası oluşturmaktadır. R2 direnci ile T8 transistörünün kollektöründen akan akımla ters orantılı bir gerilim elde edilmektedir. Bu gerilim T1, T3 ve T5 transistörlerinin bazına uygulanarak akım karşılaştırması yapılmaktadır. T1-T2, T3-T4 ve T5-T6 transistör çiftleri de sırasıyla $2.5I_0$, $1.5I_0$ ve $0.5I_0$ akım seviyeleri için akım karşılaştırıcı görevi görmektedir. Yani $X_i(0,1,2,3)$ giriş işareti için T2 üzerinden akacak akım $(I_0, I_0, I_0, 0)$, T4 üzerinden akacak akım $(I_0, I_0, 0, 0)$ ve T6 üzerinden akacak akım da $(I_0, 0, 0, 0)$ olacaktır. Bu transistörlerin kollektörleri de birleştirilerek X_i giriş akımının tersi elde edilmiştir. R4 ve T12 de I_0 referans akımını elde etmek için kullanılmıştır. T9, T10 ve T11

transistorları da elde edilen referans akımını aynalamak için devreye konulmuştur. R5-8 ile de akım karşılaştırıcılarda kullanılmak üzere referans gerilimleri elde edilmektedir. Akımı gerilime veya tersi işlem dikkat edilirse ECL teknolojisinde dirençler ile yapılmaktadır.

1.2.3. CCD teknolojisi (Charge Coupled Devices)

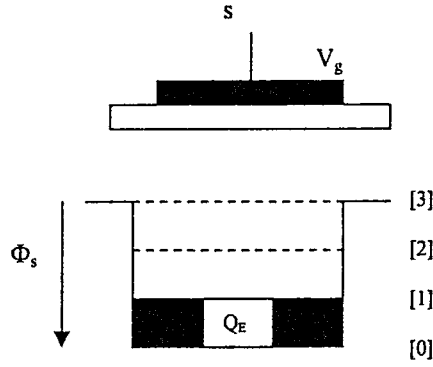
CCD adından anlaşılacağı üzere yük modlu çalışan devrelerdir. Daha evvel ikili sayı sistemini kullanan devrelerde de kullanılmıştır. CCD yük miktarının kuantalanması ve düşük güç tüketimi ve geniş çapta tümleştirmeye elverişli olduğundan ilgi görmektedir. CCD 1970 yılında Boyle ve Smith tarafından geliştirilmiştir [15]. Temel yapıya bakacak olursak seri MOS kapasitelerden oluşuyormuş gibi düşünülebilir.



Şekil 1.7 CCD elemanı

Temel olarak bir CCD üç kısımdan oluşur; giriş, transfer ve çıkış. Girişte gerilim yüke çevrilir, çıkışta ise tam tersi işlem yapılır. Transfer kısmında görüleceği üzere transfer (t) ve depolama (s) kapıları bulunmaktadır.

Depolama kapısına V_g gerilimi uygulanırsa, silisyum-silisyumdioksit ara yüzeyindeki potansiyel Φ_s siyah koyu çizgilerle belirtildiği gibi olur (Şekil 1.8). Bu potansiyel kuyusu belirli miktarda yük depolayabilir.



Şekil 1.8 CCD yük kuyusu

ÇSM için bu kuyu yükü aşağıdaki gibi ifade edilmektedir;

$$Q_s = r \cdot Q_E \quad (1.2)$$

Burada r kullanılan sayı tabanını belirtmektedir. Q_E de yük paketi adını almaktadır. Bunun değeri yaklaşık olarak 0,1-0,3pC civarındadır [16].

1.2.4. MOS teknolojisi

ÇSM ile tasarlanmış ilk CMOS akım modlu devre 1983 yılında literatüre yansımıştır. MOS teknolojisi ile hem akım modlu hem de gerilim modlu ÇSM devreleri tasarlanmaktadır. Özellikle gerilim modlu devre sayısı çoğunluktadır. Akım modlu çalışan devrelerde Bipolar teknolojisi baskındır. CMOS teknolojisinin tasarımlarda tercih edilmeyişinin başlıca sebepleri şunlardır; CMOS ikili mantık devreleri statik güç tüketmez, ancak bu durum CMOS ÇSM devreleri için geçerli değildir. Diğer önemli bir sorun da ikili CMOS devrelerinden farklı olarak ÇSM CMOS devreleri kendinden-yenilemeli (restoratif) değildir. Birkaç ardışık devrenin ardından işareti yeniden oluşturmak için seviye yenileyici kullanmak gerekir [8]. Bu olumsuz yanlarına rağmen tümdevre teknolojisindeki gelişmelere bağlı olarak CMOS devrelere olan ilgi giderek artmaktadır. Yapılan tasarımlarda birçok işlemin CMOS teknolojisi ile akım modlu olarak ikili eşdeğerlerine göre çok daha az sayıda transistor ile gerçekleştirilebileceği gösterilmiştir. MOS teknolojisi ile devre gerçekleştirme tekniklerine dördüncü bölümde ayrıntılı olarak değinilecektir.

1.3. ÇSM Uygulama Alanları

ÇSM uygulama alanları gün geçtikçe yaygınlaşmaktadır. Günümüze kadar bazı alanlarda klasik devrelere göre önemli ilerlemeler kaydedilmiştir. Bu uygulamalar aşağıda genel hatları ile verilmiştir;

1.3.1. Bellekler

ÇSM ile yapılan prototip üretimlerin çoğunluğunu bellekler oluşturmaktadır. Bellek tipleri ve üretilen prototipler hakkında genel bilgi aşağıda verilmiştir.

1.3.1.1. ROM

ROM ÇSM'nin en başarılı uygulama alanı olduğu söylenebilir. Dao [17] tarafından ROM tasarımı için önerilen çok kollektörlü I^2L teknolojisi ile devre alanının %30 küçültülebileceği gösterilmiştir [18]. Intel tarafından 1980 yılında ticari olarak piyasaya sürülen 8087 matematik işlemcisi 4 seviyeli mantık ile tasarlanmış ROM belleği içermektedir. Bellek NMOS teknolojisi ile gerçekleştirilmiştir. İkili mantık eşdeğerine göre boyutları %30 küçültülmüştür.

ROM bellek hücrelerinden ÇSM içerisinde çok önemli bir yere sahip olan PLA (Programmable Logic Array) tasarımında da kullanılabileceği belirtilmektedir [2].

1.3.1.2. Seri bellekler

ÇSM uygulama alanlarından biri de seri belleklerdir. Seri bellek tasarımında CCD teknolojisi yaygın olarak kullanılmaktadır. 4 seviyeli mantık ile seri bellek tasarımının mümkün olabileceği gösterilmiştir [19]. Bu konuda ise prototip iki tasarım IBM ve Mitsubishi tarafından CCD teknolojisi ile üretilmiştir [5].

1.3.1.3. RAM

RAM tasarımında ÇSM devreleri ile tasarımlar yapılmıştır, ancak bu devreler pek başarılı olamamıştır. Özellikle işareti algılama ve yeniden oluşturmada ciddi sorunlar bulunmaktadır. RAM bellek hücreleri, ikili mantık ile tek MOSFET kullanılarak gerçekleştirilebilmektedir. ÇSM tasarımlarında özellikle işareti yeniden oluşturmak için

kullanılan ilave elemanlar maliyeti arttırmaktadır. Ayrıca kararlılık sorunları da önemli bir engel teşkil etmektedir [2].

1.3.2. Aritmetik uygulamalar

ÇSM, toplama ve çarpma devreleri tasarımında da kullanılmıştır. Bu aritmetik uygulamalar aşağıda genel hatları ile incelenmiştir.

1.3.2.1. Çarpma devreleri

CMOS akım modlu devrelere en iyi örnek ise 54*54 bit çarpma devresidir. Çarpma devresi “artık ikili” mimarisi ile gerçekleştirilmiştir. Devre 0,5µm CMOS teknolojisi ile üretilmiştir. Aktif alan 3,05*3,08 mm² olup tranzistor sayısı 78 800’dür. Bu devre günümüze kadar gerçekleştirilen en küçük çarpma devresidir. Devre 3,3V’ta çarpma işlemini 8,8ns’de gerçekleştirmektedir. Güç tüketimi de 100MHz’de yaklaşık 540mW’tır [20].

Daha önce gerçekleştirilen 32*32bit çarpma devresi ise işaretli sayı sistemini kullanmaktadır. İkili devrelerle birlikte kullanılabilmesi için devre çıkışında 64 bit ÇSM-ikili seviye dönüştürücü kullanılmıştır. İyi bir örnek vermesi bakımından söz konusu devre ile en iyi ikili mantık eşdeğerinin ayrıntılı karşılaştırması aşağıdaki tabloda verilmiştir [21].

Tablo 1.1 Çarpma devreleri karşılaştırması

	ÇSM ÇARPMA DEVRESİ	EN HIZLI İKİLİ ÇARPMA DEVRESİ
İşlem süresi (ns)	59	56
İç düğüm sayısı	200	1,500
Tranzistör sayısı	23,600	45,000
Etkin alan (mm ²)	5,2*3,2=16,6	5,3*5,7=30,2
Güç tüketimi (W)	0,5	1
Teknoloji	2µm CMOS ve PMOS	2µm CMOS

1.3.2.2. Toplayıcılar

ÇSM'ye uygulanabilen bütün teknolojiler ile tasarlanmış tam toplayıcı devreleri mevcuttur. CMOS tekniği ile gerçekleştirilen ve akım modlu 4'lü sayı sistemini kullanan tam toplayıcı devresi ise 1994 yılında literatüre yansımıştır. Devre 4µm CMOS tekniği ile prototip olarak üretilmiştir. Devrede akım seviyesi olarak 10µA seçilmiştir. Devre hızı ise giriş akımının seviyesine bağlı olarak 34ns ile 60ns arasında değişmektedir [22].

1.3.3. Haberleşme arabirimleri

Kodlu haberleşme sistemlerinde kullanılan MC10194 tümdevresi 1980 yılında Motorola tarafından ECL teknolojisi ile gerçekleştirilmiştir [2]. Ayrıca Motorola'nın uzaktan kumanda sistemleri için geliştirdiği MC 145026/7/8 1980'den beri CMOS teknolojisi ile üretilmektedir [8].

1.3.4. İşaret işleme

İşaret işleme alanında da ÇSM devrelerine rastlamak mümkündür. Tipik uygulamalara, Fourier dönüşümü [23] ve dijital filtreleme örnek olarak verilebilir [24]. İşaret işlemede özellikle hızları oldukça yüksek olan CCD teknolojisi ile gerçekleştirilen devrelerin ticari olarak gerçeklemeye en elverişli olduğu belirtilmektedir [8].

1.3.5. Diğer uygulamalar

ÇSM devrelerini, ikili çalışan günümüz devreleriyle uyumlu çalışır hale getirmek için ÇSM-ikili (kod çözücü) ve ikili-ÇSM (kodlayıcı) dönüştürücü devreleri tasarlanmıştır. Akım modlu çalışan ve CMOS teknolojisine uyumlu olarak tasarlanan kodlayıcı-kod çözücü devreleri de mevcuttur [25]. Diğer bir uygulama da akım modlu olarak tasarlanan analog dijital dönüştürücülerdir. CMOS tekniği ile 4 seviyeli olarak tasarlanan bir örneği [25]'te verilmiştir.

Birçok ÇDM operatörünün analizinde PLA'lar kullanılmıştır. Sasao tipi PLA oldukça başarılı olmuştur. 1991 yılında ise akım modlu ve CMOS teknolojisi ile

gerçeklenmiş PLA devresi literatüre yansımıştır. 4 değerli PLA 3 μ m CMOS teknolojisi ile prototip olarak üretilmiştir [26].



2. ÇSM CEBRİ

ÇSM henüz yeni bir araştırma alanı olduğundan kullanılan simgelerde ve sembollerde uluslararası standart oluşturulamamıştır. Hala, tasarlanan fonksiyon sembolleri ve simgeleri karmaşıklığını korumaktadır. Ayrıca kullanılan işaret seviyeleri de standartlaştırılamamıştır. Örneğin akım modlu devreler için genellikle 1 mantık seviyesine karşı düşen taban akımı olarak $5\mu A$, $10\mu A$ ve $20\mu A$ kullanılmaktadır. Ancak günümüz ikili sayı teknolojisi ile uyumluluk için devreler kodlayıcı-kodçözücü devreleri ile kullanıldığından bazı uygulamalarda akım seviyeleri sorun teşkil etmemektedir. Bu çalışmada birçok araştırmacı tarafından kullanılan belli başlı bazı notasyonlar verilecektir.

Yüksek sayı tabanı kullanan sistemlerde birim yük, akım veya gerilim tanımlanır ve bunların katları sayılara karşı düşürülür. Şöyle ki; r sayı tabanı olmak üzere bu birim değerler $0,1,2,3,\dots,(r-2),(r-1)$ 'e karşı düşürülür. Diğer yaygın notasyon da, tek tabanlı sayı sistemleri için uygulanmaktadır. $r=(2K+1)$ olmak üzere ve değerler $(-K),(1-K),\dots,-2,-1,0,1,2,\dots,(K-1)(K)$ şeklinde ifade edilmektedir. Bu notasyon şekline ayarlı veya işaretli sıralama adı verilmektedir (balanced,signed). Anlaşılacağı üzere birinci notasyon çift veya tek sayılar için kullanılabilir olmasına rağmen ikinci notasyon sadece tek sayı sistemleri için geçerlidir. Birinci notasyon için bir örnek vermek gerekirse; akım modlu çalışan bir sistemde taban akımı $10\mu A$ olsun bu durumda sayı dizimiz $0, 10\mu A, 20\mu A\dots$ şeklinde olacaktır [8].

ÇSM'de, girişleri çıkışlara bağlayan operatörler (fonksiyonlar) genel olarak ikili fonksiyonlardan türetilmiştir. Ancak tanımlanabilecek operatör sayısı ikili mantığa göre daha fazladır. Fonksiyonlar ÇSM'de iki seviyeli sistemde olduğu gibi çarpanların toplamı (sum of products) formunda gösterilmektedir. Ancak ikili mantık'tan farklı olarak aynı fonksiyonun birden çok gösterilimi mevcuttur. Ayrıca daha sonra da değinileceği üzere "çarpanları" oluşturmak için kullanılan operatör sayısı arttıkça indirgeme olasılığı da artmaktadır [4,27].

2.1. ÇSM Operatörleri

Genel olarak, r sayı tabanını kullanan bir değişkenli fonksiyonun her bir girişi için r değişik çıkış olabilir. Dolayısıyla tek değişken ile r^r değişik operatör gerçekleştirilebilir. Zaten ikili sayı sisteminden bilindiği üzere $2^2=4$ adet operatör tek değişkenle (unary) gerçekleştirilebilir. Bunlar benzerlik, tersleme ve sabit (0 ve 1) operatörleri olarak isimlendirilebilir. Daha yüksek tabanlı sayı sistemleri için durum tabloda verilmiştir.

Tablo 2.1 Sayı tabanına göre fonksiyon sayısı

r	r^r
2	4
3	27
4	256
5	3125
6	46656
7	823543
8	16777216

Tablodan görüleceği üzere tek değişkenle gerçekleştirilecek operatör sayısı, sayı tabanı arttıkça üstel olarak artmaktadır. ÇSM’de kullanılan operatörler üç grup altında verilmektedir. Birinci grup operatörler tek değişkenli (unary) operatör grubudur. Literal, Döngü v.b. operatörler bu grupta yer almaktadır. İkinci grupta iki ve çok değişkenli operatörler yer almaktadır. Üçüncü grupta ise birleştirici operatörler (BOP) yer almaktadır. Bu grupta Tsum, Sum, Modsum gibi operatörler yer almaktadır [8].

2.1.1. Tek değişken (unary) operatörleri

Tek değişken operatör devrelerinin girişine sadece bir giriş işareti uygulanabilmektedir. Bu tür devreler çarpanların toplamı olarak ifade edilen fonksiyonda genellikle çarpanın içerisinde yer almaktadır.

2.1.1.1. Literal operatörü

Literal tek değişkenli bir operatördür. Literal, giriş işaretinin belirli bir kapalı aralık ($[a,b]$) içerisinde yer alması durumunda k sayısına karşı gelen mantık seviyesini oluşturmaktadır. Aralığın dışında kalan giriş değerleri için ise 0 çıkışı vermektedir. Sayı aralığında $a=b$ seçilirse giriş işaretinin sadece bir noktasında çıkış vermektedir. Bu özelliğinden dolayı Literal ile her tür fonksiyon gerçekleştirilebilmektedir. ÇSM en çok kullanılan operatördür. Bir x sayısının Literal'i aşağıdaki gibi tanımlanmaktadır [7];

$$k[a\{x\}^b] = \begin{cases} k & a \leq x \leq b \\ 0 & \text{aksi halde} \end{cases} \quad (2.1)$$

burada;

$$a\{x\}^b = \begin{cases} 1 & a \leq x \leq b \\ 0 & \text{aksi halde} \end{cases} \quad (2.2)$$

Denklem (2.2)'deki ifadeye iç Literal veya pencere Literali adı verilmektedir. $a, b \in r$ ve k da Literalin değeri olarak adlandırılmaktadır.

Literal için aşağıdaki yalın notasyon da kullanılmaktadır.

$$x = \langle 0123 \rangle \quad 3[1\{x\}^2] = \langle 0330 \rangle \text{ vb.}$$

2.1.1.2. Eşlenik literal operatörü

Eşlenik Literal operatörü aşağıdaki gibi tanımlanmıştır [28];

$$k[\overline{a\{x\}^b}] = \begin{cases} 0 & a \leq x \leq b \\ k & \text{aksi halde} \end{cases} \quad (2.3)$$

$$\text{Örnek; } x = \langle 0123 \rangle \quad 1[\overline{1\{x\}^2}] = \langle 1001 \rangle$$

$r=4$ için bazı Literallerin/eşlenik Literallerin doğruluk tablosu aşağıda verilmiştir[1];

Tablo 2.2 Bazı Literallerin doğruluk tablosu

x	$3\left[{}^1\{x\}^2\right]$	$2\left[{}^3\{x\}^3\right]$	$3\left[{}^1\{x\}^3\right]$	$1\left[{}^1\{x\}^2\right]$
0	0	0	3	1
1	3	0	0	0
2	3	0	0	0
3	0	2	0	1

2.1.1.3. Saat yönü döngü operatörü (Clockwise Cycle Operator - CWC)

CWC operatörü ÇSM işaretlerine belirli bir işaret seviyesi ekleyip modülünü alarak ötelemektedir. Tanım ifadesi aşağıda verilmiştir [29];

$$x \xrightarrow{m} = (x + m) \bmod r \quad (2.4)$$

İfadede $m \in r$ 'dir. (2.4) ifadesinden görüldüğü üzere x işaretine m işareti eklenmekte, toplamın r-1 değerini aşmaması için r'e göre modülü alınmaktadır.

Örnek; $m=2$ ve $x=1$ için $x \xrightarrow{m} = 3$, $x=2$ için $x \xrightarrow{m} = 1$ olacaktır.

2.1.1.4. Ters saat yönü döngü operatörü (Counter Clockwise Cycle Operator)

Ters saat yönü Döngü operatörü, saat yönü Döngü operatörüne bağlı olarak aşağıdaki gibi tanımlanmıştır [29];

$$x \xrightarrow{m} = x \xleftarrow{r-m} \quad (2.5)$$

Örnek; $r=4$ ve $x=<0123>$ için

$$x \xrightarrow{0} = <0123>, \quad x \xrightarrow{1} = <1230>, \quad x \xrightarrow{2} = <2301>, \quad x \xrightarrow{3} = <3012>$$

$$x \xleftarrow{1} = <3012>, \quad x \xleftarrow{2} = <2301>, \quad x \xleftarrow{3} = <1230>$$

2.1.2. İki ve çok değişkenli operatörler

Bu kısımda iki veya ikiden çok değişkenin en küçüğünü çıkış olarak veren Min ve iki veya ikiden çok değişkenin aşağıda belirtilen koşullar altında farkını alan Tdif operatörü incelenmiştir.

2.1.2.1. Min operatörü

Min operatörü giriş işaretlerinden küçük olanını çıkışa iletmektedir. Tanım ifadesi aşağıda verilmiştir [7,16];

$$\text{Min}(x_1, x_2, \dots, x_n) = x_1 \bullet x_2 \bullet \dots \bullet x_n \quad (2.6)$$

Veya;

$$\text{Min}(x_1, x_2) = \begin{cases} x_1 & x_1 \leq x_2 \text{ ise} \\ x_2 & \text{aksi halde} \end{cases}$$

Örnek; $x_1=5$ $x_2=3$ $x_3=2$ için $\text{Min}(5,3,2)=2$

Min operatörünün ikili karşılığı AND operatörüdür ve $\bullet$ sembolü ile gösterilmektedir [4]. Dört seviyeli iki değişkenli fonksiyon için Min işleminin doğruluk tablosu aşağıda verilmiştir;

Tablo 2.3 Min operatörünün doğruluk tablosu

x_1	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
x_2	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
$\text{Min}(x_1, x_2)$	0	0	0	0	0	1	1	1	0	1	2	2	0	1	2	3

2.1.2.2. Sınırlı fark (Truncated Difference-Tdif) operatörü

Tdif operatörü, bir giriş diğer bir girişten büyük olduğu sürece iki girişin farkını almaktadır ve $\boxminus$ sembolü ile gösterilmektedir. Tdif operatörünün tanımı aşağıda verilmiştir [32];

$$x \ominus y = \begin{cases} x - y & x \geq y \text{ ise} \\ 0 & \text{aksi halde} \end{cases} \quad (2.7)$$

Dört seviyeli iki değişkenin Tdif doğruluk tablosu aşağıda verilmiştir;

Tablo 2.4 Sınırlı fark operatörünün doğruluk tablosu

x	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
y	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
$x \ominus y$	0	0	0	0	1	0	0	0	2	1	0	0	3	2	1	0

2.1.3. Birleştirici operatörler

Bu bölümde “çarpanların toplamı” gösteriliminde mantık toplama işlemini yerine getiren operatörler incelenecektir.

2.1.3.1. Max veya Sum operatörü

Max operatörü iki işareten büyük olanını çıkışa iletmektedir. Tanım ifadesi aşağıda verilmiştir [4,16];

$$\text{Max}(x_1, x_2) = (x_1 \cup x_2) = \begin{cases} x_1 & x_1 \geq x_2 \text{ ise} \\ x_2 & \text{aksi halde} \end{cases} \quad (2.8)$$

Sum operatörünün ikili karşılığı OR işlemidir ve genellikle $\cup$ sembolü ile verilmektedir. İlk ÇSM tasarımlarında birleştirici operatör olarak sadece Max kullanılmıştır. Dört seviyeli ve iki değişkenli Sum işleminin doğruluk tablosu aşağıda verilmiştir;

Tablo 2.5 Max operatörünün doğruluk tablosu

x_1	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
x_2	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
$\text{Max}(x_1, x_2)$	0	1	2	3	1	1	2	3	2	2	2	3	3	3	3	3

2.1.3.2. Modsum operatörü

Modsum operatörü değişkenlerin modül r toplamıdır. İkili mantık karşılığı ise EXOR'dur ve $\oplus$ sembolü ile verilmektedir [8].

$$x_1 \oplus x_2 = \text{Modsum}(x_1, x_2) = (x_1 + x_2)_{\text{mod } r} \quad (2.9)$$

Dört seviyeli iki değişkenli sistem için Modsum operatörünün doğruluk tablosu aşağıda verilmiştir;

Tablo 2.6 Modsum operatörünün doğruluk tablosu

x_1	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
x_2	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
$x_1 \oplus x_2$	0	1	2	3	1	2	3	0	2	3	0	1	3	0	1	2

2.1.3.3. Tsum (Truncated Sum) operatörü (sınırlı toplam)

Tsum operatörü, giriş işaretlerinin lineer toplamını alıp çıkışa yansıtmaktadır ve $\boxplus$ sembolü ile verilmektedir. Ancak çıkışa yansıyan toplam işaret $r-1$ işaret seviyesi ile sınırlandırılmaktadır. Tanım ifadesi aşağıda verilmiştir [33];

$$\text{Tsum}(x_1, x_2, \dots, x_n) = x_1 \boxplus x_2 \boxplus \dots \boxplus x_n = \text{Min}(x_1 + x_2 + \dots + x_n, r-1) \quad (2.10)$$

Örnek; $r=4$ için $x_1=2$, $x_2=3$ ve $x_3=1$ $\text{Tsum}(2,3,1)=\text{Min}(2+3+1, r-1)=\text{Min}(6,3)=3$

Dört seviyeli iki değişkenli fonksiyonun Tsum doğruluk tablosu aşağıda verilmiştir;

Tablo 2.7 Sınırlı toplam operatörünün doğruluk tablosu

x_1	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
x_2	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
$\text{Tsum}(x_1, x_2)$	0	1	2	3	1	1	3	3	2	2	3	3	3	3	3	3

Ayrıntılı incelenen operatörlerle birlikte diğer yaygın operatörler Tablo2.8'de verilmiştir [2];

Tablo 2.8 Yaygın olarak kullanılan ÇSM operatörleri

Sıra	Yaygın adı	Birincil	İkincil	Açıklamalı gösterim $r=4$ $a=1, b=2, y = \bar{x}$	Değer Alma koşulu	*
1	Yineleme	x		<0123>	Standart x	a
2	Tümleyen	$\bar{x}$		<3210>	$(r-1)-x$	a
3	Max	$\text{Max}(x,y)$	$x+y$ $x \cup y$	<3223>	x eğer $x \geq y$ aksi halde y	b
4	Min	$\text{Min}(x,y)$	$x \cdot y$ $x \cap y$	<0110>	x eğer $x \leq y$ aksi halde y	a
5	Successor	$\vec{x}$	$\text{suc}(x)$	<1230>	$(x+1) \bmod(r)$	b
6	Döngü (saat yönü)	$\overset{b}{x} \rightarrow$		<2301>	$(x+b) \bmod(r)$	b
7	Döngü (ters saat yönü)	$\overset{b}{x} \leftarrow$		<2301>	$(x-b) \bmod(r)$	b
8	Literal	$\left[\overset{a}{\{x\}}^b \right]$		<0330>	$(r-1)$ eğer $a \leq x \leq b$ aksi halde 0	a
9	Delta Literal J fonksiyon	$\begin{bmatrix} ab \\ x \end{bmatrix}$		<0300>	$(r-1)$ eğer $x=a$ aksi halde 0	b
10	Kapalı ara (closed interval)	$\begin{bmatrix} ab \\ x \end{bmatrix}$		<0110>	1 eğer $a \leq x \leq b$ aksi halde 0	c
11	Açık interval	$\begin{bmatrix} a & b \\ x \end{bmatrix}$		<0000>	1 eğer $a < x < b$ aksi halde 0	c
12	Delta interval	$\begin{bmatrix} a \\ x \end{bmatrix}$		<0100>	1 eğer $x=a$ 0 aksi halde	c
13	Yarı interval	$\begin{bmatrix} a \\ x \end{bmatrix}$		<0111>	1 eğer $a \leq x$ 0 aksi halde	c
14	Yarı interval	$\begin{bmatrix} a \\ x \end{bmatrix}$		<1100>	1 eğer $x \leq a$ 0 aksi halde	c
15	Yarı interval	$\begin{bmatrix} a \\ x \end{bmatrix}$		<1000>	1 eğer $x < a$ 0 aksi halde	c
16	Yarı interval	$\begin{bmatrix} a \\ x \end{bmatrix}$		<0011>	1 eğer $a < x$ 0 aksi halde	c
17	Threshold Literali (yukarı)	$U_a(x)$		<0111>	1 eğer $x \geq a$ 0 aksi halde	c
18	Adım Literali (aşağı)	$D_a(x)$		<1100>	1 eğer $x \leq a$ 0 aksi halde	c
19	Sınırlı Fark	$x \ominus a$		<0012>	$x-a$ eğer $x \geq a$ 0 aksi halde	b
20	Sınırlı toplam	$x \boxplus a$		<1233>	$x+a$ eğer $\leq r-1$, aksi halde $r-1$	a
21	T kapısı	$T(x,y,q)$		<0123>	x eğer $q=0$ y eğer $q=1$	c
22	ÇSMNOR	$\overline{\text{Max}}(x,y)$	$\text{Min}(\bar{x}, \bar{y})$	<0110>	$\bar{x}$ eğer $x \geq y$, $\bar{y}$ aksi halde	b
23	ÇSMNAND	$\overline{\text{Min}}(x,y)$	$\text{Max}(\bar{x}, \bar{y})$	<3223>	$\bar{x}$ eğer $x \leq y$, $\bar{y}$ aksi halde	b

Tablo 2.8’de * sütununda operatörlerin CMOS akım modlu tasarımlarda kullanım sıklığına göre kodlama yapılmıştır. Kodlama açılımı aşağıda verilmiştir;

- a. Çok kullanılıyor.
- b. Nadiren kullanılıyor.
- c. CMOS olarak tasarımı bulunmuyor.

2.2. ÇSM Fonksiyonlarının Cebirsel İfade Yöntemleri

ÇSM fonksiyonları klasik ikili mantık cebirinde de kullanılan çarpanların toplamı formunda verilmektedir. Çarpanlar genellikle Min operatörü ile birleştirilmiş tek değişkenli (unary) operatörlerden oluşmaktadır. Çarpanlar da birleştirici operatörler ile birleştirilerek bütün fonksiyon oluşturulmaktadır. ÇSM tasarımlarının ilk yıllarında fonksiyonlar polinom şeklinde gösterilmiştir. Polinomdaki terimler üzerinde yapılan işlemlerle fonksiyon klasik çarpanların toplamı formuna dönüştürülebilmektedir. Bu ifade yöntemlerinin temel üç tanesi aşağıda verilmiştir.

2.2.1. ÇSM fonksiyonlarının polinom gösterimi

Herhangi bir ÇSM fonksiyonu aşağıdaki gibi ifade edilebilir [34];

$$f(x) = \sum_{i=0}^{r-1} c_i x^i \quad (2.11)$$

İfadedeki c_i katsayıları Menger teoreminden yararlanılarak bulunmaktadır [35].

$c_0 = f_0$ olmak üzere;

$$c_i = \sum_{g=0}^{i-1} [f(0) - f(g)] g^{-i} \quad (2.12)$$

Verilen bir fonksiyonun $0,1,...,t$ noktalarında $f_0, f_1, ..., f_t$ değerlerini alsın. Bu fonksiyonun c katsayıları aşağıdaki lineer denklemlerden elde edilebilir;

$$c_0 + c_1 0 + c_2 0^2 + \dots + c_t 0^t = f_0$$

$$c_0 + c_1 1 + c_2 1^2 + \dots + c_t 1^t = f_1 \quad (2.13)$$

.....

$$c_0 + c_1 t + c_2 t^2 + \dots + c_t t^t = f_t$$

Yukarıdaki (2.12) eşitliği kullanılarak elde edilen $f_t - f_0$ ifadesi bütün denklemlerden çıkarılırsa aşağıdaki eşitlik takımı elde edilir;

$$(2-1)c_1 + (2^2-1^2)c_2 + \dots + (2^t-1^t)c_t = f_2 - f_1$$

$$(3-1)c_1 + (3^2-1^2)c_2 + \dots + (3^t-1^t)c_t = f_3 - f_1 \quad (2.14)$$

.....

$$(t-1)c_1 + (t^2-1^2)c_2 + \dots + (t^t-1^t)c_t = f_t - f_1$$

Sonlu uzayda ($0 < n < t-1$) olmak üzere;

$$(a \pm b)^n = a^n \pm b^n \quad (2.15)$$

(2.15) ifadesinin doğru olduğu varsayımıyla denklem takımındaki her $f_j - f_i$ 'i içeren denklemler $j-i$ 'ye bölünebilir. Buradan hareketle;

$$c_1 + (2-1)c_2 + \dots + (2-1)^{t-1}c_t = \frac{f_2 - f_1}{2-1}$$

$$c_1 + (3-1)c_2 + \dots + (3-1)^{t-1}c_t = \frac{f_3 - f_1}{3-1} \quad (2.16)$$

.....

$$c_1 + (t-1)c_2 + \dots + (t-1)^{t-1}c_t = \frac{f_t - f_1}{t-1}$$

Denklem takımları birbirinden çıkartılarak bilinmeyen birini elimine edilebilir. Sonlu uzay varsayımının doğru olduğu kabulü ile çıkarma işlemi sol tarafta bir bilinmeyen kalana kadar uygulanırsa ve (2.12) $i = k - 1$ konarak Menger formülü elde edilir;

$$c_{k-1} = -f_0 - f_1, \dots, -f_{k-1} \quad (2.17)$$

Bu katsayılar Newtonun sınırlı farklar yöntemi ile de elde edilebilir. Bu durumda fonksiyonumuz aşağıdaki eşitlikteki gibi elde edilir;

$$f(x) = f(x_0) + \sum_{i=1}^{r-1} (x - x_0)(x - x_1) \dots (x - x_{i-1}) [x_0, x_1, \dots, x_i] \quad (2.18)$$

Yukarıdaki ifade tek değişkenli bir ÇSM fonksiyonu için verilmiştir. İfade n değişkenli fonksiyonlar için genişletilebilir. İfadenin genel hali aşağıdaki gibi olmaktadır;

$$f(x_1, x_2, \dots, x_n) = \sum_{i=0}^{r-1} c_i \prod_{j=1}^n x_j^{i_j} \quad (2.19)$$

Fonksiyondaki katsayıları elde etmek için r^n adet denklemin oluşturulması ve çözülmesi gerekmektedir. İki değişkenli bir fonksiyonun katsayıları aşağıdaki yol izlenerek çözülebilir;

$$f(x_1, x_2) = \sum_{i=0}^{r-1} \sum_{j=0}^{r-1} c_{ij} x_1^i x_2^j \quad (2.20)$$

Burada $f_{ij} : 0 < i, j < r$ katsayıları matris oluşturmaktadır. Öncelikle x_2 'nin sıfır olduğunu varsayalım. Bu durumda $f_{00}, f_{10}, f_{20}, \dots, f_{(r-1)0}$ 'dan oluşan denklem takımları oluşturulur ve $c_{00}, c_{10}, \dots, c_{(r-1)0}$ katsayıları hesaplanır. Bu eşitlikler yukarıda verilen bir değişkenli eşitliklere dönüşmüş olur. Ayrıca simetrik durum düşünülerek x_1 'in de sıfır olduğu varsayılırsa bu durumda c_{00} bulunmuş olur. Geri kalan katsayıları bulmak için ayrıştırma (decomposition) yaklaşımı kullanılabilir. Şöyle ki x_1 'i içeren $r-1$ tane eşitliği oluşturmak için x_2 değişkenine $1, 2, \dots, r-1$ değerleri atansın;

$$f(x_1, i) = D_{0i} + D_{1i}x_1 + \dots + D_{(r-1)i}x_1^{r-1} \quad (2.21)$$

D katsayıları yukarıda verilen bir değişkenli sistem denklemlerinden hareketle çözülebilir. Aşağıdaki eşitlik yardımı ile de geri kalan r-1 tane katsayı hesaplanabilir [35];

$$D_{1i} = c_{10} + i c_{11} + \dots + i^{r-1} c_{1(r-1)} \quad (2.22)$$

Bu ifade yöntemi özellikle belirsiz nokta sayısı fazla olan fonksiyonlar (iyi tanımlanmamış) için uygundur. Burada polinomun derecesi fonksiyonun *belirli* (fonksiyonda sıfırdan farklı nokta) nokta sayısı kadar olduğundan *belirsizlik* noktalarını elemek için polinom gösterimi kullanılmaktadır. İki den fazla değişkenli fonksiyonların katsayı hesaplamaları için [30]'a bakılabilir.

2.2.2. Reed-Müller kanonik formları

Yukarıda verilen polinom gösteriliminden hareketle iki değişik polinom gösterilimi önerilmiştir. Bu iki ifade yöntemi arasında temelde fark bulunmamaktadır. Sadece kullanılan operatörler farklıdır. Bu iki ifade yönteminin temeli ikili fonksiyonlardan türetilmiş çarpanların toplamı olarak bilinen genelleştirilmiş Reed-Müller veya Tamari açılımına dayanmaktadır [5].

2.2.2.1. Max kanonik formu

Sadece Döngü, Min ve Sum veya Max operatörleri kullanılarak herhangi bir ÇSM fonksiyonu aşağıdaki kanonik formda ifade edilebilir.

$$f(x_1, \dots, x_n) = \sum_{k=1}^{R-1} \left\{ \prod_{V|f(V)=k} \left(\sum_{i=1}^n x_i^{v_i} \right) \right\}^k \quad (2.23)$$

$\sum \equiv$ mantık toplama işlemi, $\prod \equiv$ mantık çarpım işareti (Min operasyonu).

$V|f(V)=k$, $V=v_1, v_2$, 'de fonksiyonun k'ya eşit olduğu girişler kümesidir. Bu ifadeye toplamaların çarpımının toplamı adı verilmektedir.

Tablo 2.9 İki değişkenli dört seviyeli fonksiyon

x_1	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
x_2	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
$f(x_1, x_2)$	2	2	2	3	1	2	3	0	1	0	0	0	2	1	1	0

Örneğin yukarıda doğruluk tablosu verilen fonksiyonun kanonik formu aşağıdaki ifadedeki gibi olacaktır [29].

$$\begin{aligned}
 f(x_1, x_2) = & \left[(x_1^{\leftarrow} \oplus x_2) \bullet (x_1^{\leftarrow 2} \oplus x_2) \bullet (x_1^{\leftarrow 3} \oplus x_2^{\leftarrow}) \bullet (x_1^{\leftarrow} \oplus x_2^{\leftarrow 2}) \right]^1 \\
 & + \left[(x_1 \oplus x_2) \bullet (x_1 \oplus x_2^{\leftarrow}) \bullet (x_1 \oplus x_2^{\leftarrow 2}) \bullet (x_1^{\leftarrow} \oplus x_2^{\leftarrow}) \bullet (x_1^{\leftarrow 3} \oplus x_2) \right]^2 \\
 & + \left[(x_1 \oplus x_2^{\leftarrow 3}) \bullet (x_1^{\leftarrow} \oplus x_2^{\leftarrow 2}) \right]^3
 \end{aligned} \tag{2.24}$$

Döngü operatörünün bazı özellikleri aşağıda verilmiştir;

$$\bullet \quad x^{\leftarrow m} = x^{\rightarrow r-m} \tag{2.25}$$

$$\bullet \quad x^{\rightarrow r} = x^{\rightarrow 0} = x \tag{2.26}$$

$\bullet \quad 1 \leq k \leq (r-1)$ ve $1 \leq m \leq (r-1)$ olmak üzere;

$$(x \bullet x^k) = 0 \tag{2.27}$$

$$(x \bullet x^k)^m = m \tag{2.28}$$

$$(x \bullet x^{\rightarrow} \bullet x^{\leftarrow 2} \bullet \dots \bullet x^{\leftarrow r-1}) = 0 \tag{2.29}$$

- $1 \leq l \leq (r - m)$ olmak üzere;

$$f \supseteq f^{\leftarrow 1} \oplus f^k \oplus f^{\leftarrow r-1} \quad (2.30)$$

- $1 \leq j \leq k$ olmak üzere De Morgan kuralından hareketle;

$$f^j \oplus f^k = f^k \quad (2.31)$$

$$f^j \bullet f^k = f^j \quad (2.32)$$

$$f_1^j \bullet f_2^j = f_1^k \bullet f_2^j = f_1^j \bullet f_2^k \quad (2.33)$$

$$(f \oplus f^{\leftarrow m})^k = 0 \quad (2.34)$$

$$f^1 \oplus f^{\leftarrow 2} \oplus f^{\leftarrow 3} + \dots \oplus f^{\leftarrow (r-2)} \oplus f^{\leftarrow (r-1)} = f^{\leftarrow r} \quad (2.35)$$

Toplamların çarpımının toplamı olarak elde edilen fonksiyona yukarıdaki özellikler uygulanırsa fonksiyonu aşağıdaki gibi çarpımların toplamı olarak basitleştirilebilir;

$$f(x_1, x_2) = x_1^{\leftarrow} \bullet x_2^2 \oplus x_1^{\leftarrow 2} \bullet x_2^{\leftarrow} \oplus x_1^3 \bullet x_2 \quad (2.36)$$

2.2.2.2. Tsum kanonik formu

Diğer bir kanonik gösterilim de Literal, Min ve Tsum operatörü ile verilen aşağıdaki ifadedir [28];

$$f = \sum_k c_k \bullet [a_{k1} \{x_1\}^{b_{k1}}] \bullet \dots \bullet [a_{kn} \{x_n\}^{b_{kn}}] \quad (2.37)$$

burada $c_n \in (1, 2, \dots, n)$.

Örnek bir fonksiyon Tablo 2.10'da verilmiştir.

Tablo 2.10 İki değişkenli örnek fonksiyon

x ₁ \ x ₂	0	1	2	3
	0	1	2	3
0	2	0	0	2
1	0	0	0	0
2	0	0	0	0
3	2	0	0	2

Tablo 2.10'da verilen fonksiyonun Tsum kanonik formu (2.38) ifadesinde verilmiştir;

$$f = 2 \cdot [^0\{x_1\}^0] \cdot [^0\{x_2\}^0] \boxplus 2 \cdot [^0\{x_1\}^0] \cdot [^3\{x_2\}^3] \boxplus 2 \cdot [^3\{x_1\}^3] \cdot [^0\{x_2\}^0] \boxplus 2 \cdot [^3\{x_1\}^3] \cdot [^3\{x_2\}^3] \quad (2.38)$$

2.2.2.3. Genelleştirilmiş Reed-Müller kanonik formu

Yukarıda tanıtılan ki genelleştirilmiş Reed-Müller açılımı birleştirilmiş ve Literal, Döngü operatörleri de dahil daha fazla operatör kullanımına izin veren aşağıdaki genel ifade verilmektedir;

$$f(x_1, x_2, \dots, x_n) = P_i \langle BOP \rangle, \dots, \langle BOP \rangle P_i \quad (2.39)$$

Bu ifadede herhangi bir birleştirici operatör kullanılabilir (BOP). ÇSM için tanımlanan birleştirici operatörler Max, Tsum ve Modsum'dur. Dolayısıyla bir ÇSM fonksiyonu birçok değişik operatör set'i ile birden çok değişik yolla gerçekleştirilebilir [33]. Bu konudaki önemli çalışmalara örnek olarak; Min, Max ve pencere Literalini kullanan Allen ve Givone [7], Min, Max, Döngü ve inverter operatörlerini kullanan Vranesic [29], Min, Tsum ve pencere Literalini kullanarak tasarım yapan McCluskey [36] verilebilir.

Yukarıdaki ifadede P sembolü ile verilen terimler çarpan (product term PT) olarak adlandırılmaktadır. Bir çarpan genel olarak Literal, eşlenik Literal, Döngü ve benzeri tek değişkenli (unary) operatörlerin Min operasyonudur. Bir çarpanda her giriş değişkeni bir kez görülür. Genel ifadesi aşağıdaki gibidir [27];

$$P = [^{a_1}\{x\}^{b_1}] \bullet \dots \bullet [^{a_n}\{x_n\}^{b_n}] = \min([^{a_1}\{x_1\}^{b_1}], \dots, [^{a_n}\{x_n\}^{b_n}]) \quad (2.40)$$

İki değişkenli ve dört seviyeli fonksiyonlar için PT örnekleri aşağıda verilmiştir;

$$1[^1\{x_1\}^1] \bullet 1[^2\{x_2\}^3], \quad 1[^1\{x_1\}^1] \bullet 2[^2\{x_2\}^3], \quad 2[^1\{x_1\}^1] \bullet 1[^2\{x_2\}^3], \quad 2[^1\{\overline{x_1}\}^1] \bullet x_2^2, \\ (x_1^2) \bullet (x_2^3), \quad 2[^1\{x_1\}^3] \text{ ve } (x_2^1).$$

Bir PT'ni ifade etmek için birçok değişik yöntem mevcuttur. Örneğin yukarıdaki örnek PT'lerin bazıları özdeşdir;

$$1[^1\{x_1\}^1] \bullet 1[^2\{x_2\}^3] = 1[^1\{x_1\}^1] \bullet 2[^2\{x_2\}^3] = 2[^1\{x_1\}^1] \bullet 1[^2\{x_2\}^3]$$

3. İNDİRGEME YÖNTEMLERİ

ÇSM fonksiyonlarının daha az çarpan (product term) ile ifade edilmesine “indirgeme” adı verilmektedir. ÇSM fonksiyonlarının indirgenmesi üzerine oldukça çok sayıda yöntem önerilmiştir [4,7,9-12,28,32,33]. Ancak bunların hiçbiri kesin sonuç vermemektedir. Bu nedenle mevcut indirgeme yöntemlerine “yaklaşık minimum” indirgeme yöntemleri adı verilmektedir [4,11].

Mevcut indirgeme yöntemleri kesin sonuç vermemekle birlikte hesaplamaları da oldukça karmaşık olduğundan el yordamı ile çözüme uygun düşmemektedir.

Tasarım için kullanılan operatörlere bağlı olarak bir ÇSM fonksiyonunu birden fazla ifade etme olanağı vardır. Bunun sonucu olarak da indirgeme yöntemleri her fonksiyon gösterilimine uygulanamamaktadır. Mesela bir ÇSM fonksiyonu Literal/eşlenik Literal veya Döngü/eşlenik Döngü operatörlerinden herhangi bir grubun veya her iki grubun kullanılmasıyla gerçekleştirilebilir. İndirgeme yöntemi üzerinde en önemli etkiye sahip faktör ise birleştirici operatörlerdir. Her birleştirici operatöre göre indirgeme yöntemi değişik olmaktadır. Bunun yanı sıra birleştirici operatörler de her teknolojiye gerçekleştirilemediğinden indirgeme yöntemleri de teknolojiye bağımlı olmaktadır. Örneğin işaretlerin lineer toplanması ile elde edilen sum işlemi akım modlu devrelerle kolayca gerçekleştirilebilirken gerilim modlu devrelerde aynı durum söz konusu değildir [2,4].

Akım modlu devrelerin indirgenmesi için verilen bir yöntemde ise, tasarım için kullanılan fonksiyonların artması ile, fonksiyon tasarım esnekliğinin arttığı ve dolayısıyla indirgeme olasılığının arttığı gösterilmiştir [33].

3.1. İndirgeme Üzerine Bazı Tanımlar

İndirgeme yöntemlerinde kullanılan bazı kavramlar aşağıdaki alt bölümlerde ayrıntılı olarak incelenmiştir.

3.1.1. Minterm

Herhangi bir $f(x_1, x_2, \dots, x_n)$ fonksiyonu için, değişkenlerin aldığı değerler aşağıdaki koşulu sağlarsa minterm olarak adlandırılır [4,33];

$$f(x_1, x_2, \dots, x_n) \neq 0 \quad (3.1)$$

İfadedeki $x_i \in (0,1,\dots,r-1)$ 'tir. Minterm çarpanın özel bir halidir ve bütün girişlere bağlıdır.

Tablo 3.1 İki değişkenli dört değerli fonksiyonun doğruluk tablosu

$x_2 \backslash x_1$	$f(x_1, x_2)$			
	0	1	2	3
0	2	d	0	0
1	2	3	1	1
2	0	0	1	0
3	3	1	0	1

Örneğin yukarıdaki tabloda beş tane 1 seviyeli, ikişer tane de 2 ve 3 seviyeli minterm vardır.

3.1.2. Keyfi değer (don't care)

Eğer bir mintermin değeri r 'ye eşitse keyfi değer olarak adlandırılır ve “d” sembolü ile gösterilir [28,33].

3.1.3. Mintermlerin kapsanması (cover)

Eğer bir mintermin değeri, keyfi değer d veya $m \in (0,1,\dots,r-1)$ olmak üzere m 'e eşitse m seviyeli bir çarpan tarafından kapsanabilir. Örneğin yukarıdaki doğruluk tablosunda $2^{[0\{x_1\}^1]} \cdot 2^{[0\{x_2\}^1]}$ çarpanı, $2^{[0\{x_1\}^0]} \cdot 2^{[0\{x_2\}^0]}$, $2^{[1\{x_1\}^1]} \cdot 2^{[1\{x_2\}^1]}$, $2^{[1\{x_1\}^1]} \cdot 4^{[0\{x_2\}^0]}$ mintermlerini kapsamaktadır [33].

3.1.4. Bileşen (implicant)

Bir $f(x_1, x_2, \dots, x_n)$ fonksiyonunun çarpanı $I(x_1, x_2, \dots, x_n)$ aşağıdaki koşulu bütün x 'ler için sağlarsa “bileşen” adını alır [4,33];

$$f(x_1, x_2, \dots, x_n) \geq I(x_1, x_2, \dots, x_n) \quad (3.2)$$

Yukarıdaki örnek fonksiyonumuz için bileşenlerin bazıları $2[{}^0\{x_1\}^1] \bullet 2[{}^1\{x_2\}^1]$, $1[{}^0\{x_1\}^1] \bullet 1[{}^1\{x_2\}^1]$, $1[{}^0\{x_1\}^1] \bullet 1[{}^2\{x_2\}^2]$ ve $1[{}^1\{x_2\}^1]$ 'dir.

3.1.5. Asal bileşen (prime implicant)

Bir $f(x_1, x_2, \dots, x_n)$ fonksiyonunun bileşeni $I(x_1, x_2, \dots, x_n)$, aşağıdaki eşitliği bütün x_i 'ler için sağlayan başka bir $I'(x_1, x_2, \dots, x_n)$ bileşeni yoksa “asal bileşen” adını alır [4,33].

$$I'(x_1, x_2, \dots, x_n) \geq I(x_1, x_2, \dots, x_n) \quad (3.3)$$

Örneğimizdeki fonksiyonun asal bileşenleri $1[{}^1\{x_2\}^1]$, $2[{}^0\{x_1\}^1] \bullet 2[{}^0\{x_2\}^1]$, $1[{}^0\{x_1\}^1] \bullet 1[{}^2\{x\}^2]$ ve'dır.

3.2. Asal Bileşenlerle İndirgeme

Bilindiği üzere ikili devrelerin indirgenmesinde asal bileşenler yeterli olmaktadır. Bundan hareketle önerilen ilk ÇSM indirgeme yöntemleri sadece asal bileşenleri dikkate almaktadır.

3.2.1. Allen ve Givone indirgeme yöntemi

Çarpanların toplamı olarak ifade edilen bir ÇSM fonksiyonunun indirgenmesine yönelik ilk yöntem Allen ve Givone tarafından 1968'de verilmiştir [7]. Bu yöntemde fonksiyonun tablo gösterilimi verilmiş ve tablodaki her satır ayrı bir çarpan ile ifade edilmiştir. Tablo 3.2'de verilen örnek fonksiyonun cebrik ifadesi (3.4)'te verilmiştir.

Tablo 3.2 İki değişkenli fonksiyonun tablo gösterilimi

x_1	x_2	$f(x_1, x_2)$
0	0	2
0	1	2
0	2	0
1	0	1
1	1	2
1	2	2
2	0	0
2	1	2
2	2	2

$$\begin{aligned}
 f(x_1, x_2) = & 2 \bullet [^0\{x_1\}^0] \bullet [^0\{x_2\}^0] + 2 \bullet [^0\{x_1\}^0] \bullet [^1\{x_2\}^1] + 1 \bullet [^1\{x_1\}^1] \bullet [^0\{x_2\}^0] \\
 & + 2 \bullet [^1\{x_1\}^1] \bullet [^1\{x_2\}^1] + 2 \bullet [^1\{x_1\}^1] \bullet [^2\{x_2\}^2] + 2 \bullet [^2\{x_1\}^2] \bullet [^1\{x_2\}^1] \\
 & + 2 \bullet [^2\{x_1\}^2] \bullet [^2\{x_2\}^2]
 \end{aligned} \tag{3.4}$$

3.2.1.1. Konsensüs (birliktelik) işlemi (*)

$P_1 = c_1 \bullet [^{a1}\{x_1\}^{b1}] \bullet \dots \bullet [^{bn}\{x_n\}^{bn}]$ ve $P_2 = c_2 \bullet [^{c1}\{x_1\}^{d1}] \bullet \dots \bullet [^{cn}\{x_n\}^{dn}]$ gibi iki çarpanın “birlikteliği” aşağıdaki gibi verilmektedir;

$$P_1 * P_2 = c_3 \bullet [^{e1}\{x_1\}^{f1}] \bullet [^{e2}\{x_2\}^{f2}] \bullet \dots \bullet [^{en}\{x_n\}^{fn}] \tag{3.5}$$

Burada $c_3 = c_1 \bullet c_2$ ve $i=k$ için;

$$[^{ek}\{x_n\}^{fk}] = [^{ak}\{x_k\}^{bk}] + [^{ck}\{x_k\}^{dk}] \tag{3.6}$$

ve $i \neq k$ için;

$$[^{ek}\{x_n\}^{fk}] = [^{ak}\{x_k\}^{bk}] \bullet [^{ck}\{x_k\}^{dk}] \tag{3.7}$$

olarak verilmektedir.

Ancak çarpanların herhangi birinde değişkenlerden biri eksik ise konsensüs çarpanının da o değişkeni içeren terimi ifade edilememektedir. Bu sorunu çözmek için eksik değişken keyfi değer'li $\left[{}^0\{x_i\}^{r-1} \right]$ olarak çarpana ilave edilmektedir.

Bir ÇSM fonksiyonunun sadece bileşenleri oluşturan indirgeme yönteminin algoritması ise şöyledir;

- a- Fonksiyonu bütün değişkenleri içerecek şekilde çarpanların toplamı kanonik formunda ifade et. Eksik değişkenleri rastgele değer atayarak fonksiyona ilave et.
- b- Başka çarpanlar tarafından kapsanan bütün değişkenleri fonksiyondan sil.
- c- Konsensüs işlemini bütün koordinatlara uygula. Eğer bu işlem sırasında oluşan konsensüs terimi $c_i \bullet \left[{}^{ei}\{x_i\}^{fi} \right]$ başka terimleri kapsarsa, kapsadığı terimleri de fonksiyondan sil. Bu işleme çarpanlar arasında birbirini kapsayan terim kalmayıncaya kadar devam et.
- d- İfadedeki bütün keyfi değerleri sil.

3.2.2. Su ve Cheung indirgeme yöntemi

Diğer bir asal bileşenlerle (prime implicant) indirgeme yöntemi Su ve Cheung tarafından önerilmiştir [37]. Yöntemde doğruluk tablosu yerine grafik gösterilim olarak kübik gösterilim kullanılmıştır. Eşlenik Literal operatörü tanımlanmış ve indirgemedede kullanılmıştır.

Önerilen yöntemde Literaller için değişik bir yazım kullanılmıştır. $\left[{}^i\{x\}^i \right] \equiv x$ olarak verilmiştir. x tek bir noktayı temsil ettiğinden nokta Literalı olarak adlandırılmaktadır.

OR birleşik Literal, aynı değişkenin indirgenmiş Literallerinin toplamıdır. Örneğin $[^0\{x\}^2] + [^1\{x\}^2]$ ve $[^1\{x\}^1] + [^3\{x\}^3]$ OR birleşik Literallerdir. Ancak $[^1\{x\}^2] + [^3\{x\}^3]$ OR birleşik Literal değildir zira $[^1\{x\}^2] + [^3\{x\}^3] = [^1\{x\}^3]$ 'tür.

AND birleşik Literal, aynı değişkenin Literallerinin (mantık) çarpımıdır. Örneğin $[^1\{x\}^2] \bullet [^1\{x\}^1]$ ve $[^1\{x\}^1] \bullet [^3\{x\}^3]$ AND birleşik Literallerdir. Ancak $[^1\{x\}^1] \bullet [^1\{x\}^3]$ AND birleşik Literal değildir zira $[^1\{x\}^1] \bullet [^1\{x\}^3] = [^1\{x\}^1]$ 'dir.

Toplam veya çarpım içerisinde birleşik Literal varsa toplam veya çarpım da birleşiktir.

3.2.2.1. Kübik gösterilim

Çok seviyeli n tane girişli bir fonksiyon, n boyutlu r^n tane küp ile ifade edilebilir. Değişkenin aldığı değerler (mintermler) r -bit alanda ifade edilebilir. Örneğin üç seviyeli ve üç değişkenli bir fonksiyonun (0,1,2) girişi kübik formda (001)(010)(100) ile verilmektedir.

$A^* = K_a(a_{10}, a_{11}, \dots, a_{1r-1})(a_{20}, a_{21}, \dots, a_{2r-1}), \dots, (a_{n0}, a_{n1}, \dots, a_{nr-1}) = K_a a_{10}, a_{11}, \dots, a_{1r-1}$. Bu ifadede $K_a \in r$ ve $\underline{a}_i$, A^* 'nin i 'ci koordinatını ve a_{ij} 'de j 'nci biti göstermektedir. Herhangi bir fonksiyon A^* lardan oluşan bir vektörle ($\underline{A}^*$) gösterilebilir. A^* alanının i 'nci küpünü göstermektedir. * sembolü olmayan terimler K_a katsayısız terimleri ifade edecektir.

Bit seviyeli OR, AND ve eşlenik işlemleri aşağıdaki gibi verilmektedir;

$$\underline{c}_i = \underline{a}_i \bullet \underline{b}_i = \begin{cases} 1 & a_{ij} = b_{ij} \\ 0 & \text{aksi halde} \end{cases} \quad (3.8)$$

$$\underline{c}_i = \underline{a}_i + \underline{b}_i = \begin{cases} 1 & a_{ij} \neq b_{ij} \\ 0 & \text{aksi halde} \end{cases} \quad (3.9)$$

$$\underline{c}_i = \underline{a}_i = \begin{cases} 1 & a_{ij} = 0 \\ 0 & \text{aksi halde} \end{cases} \quad (3.10)$$

$$\sum \underline{A} = A^1 + A^2 + \dots + A^q \quad (3.11)$$

B^* , A^* 'yı ancak ve ancak bütün i 'ler için $K_a \leq K_b$ ve $\underline{a_i} \bullet \underline{b_i} = 0^*$ koşulları altında kapsar.

İki alanın birleşimi;

$$\underline{A}^* \cup \underline{B}^* = \underline{C}^* \quad (3.12)$$

olarak verilmektedir ve $\underline{C}^*$ $\underline{A}^*$ ve $\underline{B}^*$ 'nin birbirini kapsayan elemanları hariç tüm ortak elemanları içermektedir.

İki alanın sharp işlemi ($\#$); $A \# B$ A 'nın B 'den farklı elemanlarından oluşmaktadır.

İki alanın kesişimi ($\cap$);

$$\underline{A}^* \cap \underline{B}^* = \underline{C}^* \quad (3.13)$$

olarak tanımlanmıştır. Burada $K_c = K_a \bullet K_b$ ve $\underline{c_i} = \underline{a_i} \bullet \underline{b_i}$ 'dir. $\underline{A} \cap \underline{B}$ 'de benzer olarak tanımlanmıştır.

Silme işlemi ($-$); $\underline{A} - A^i = A^i$ kübü silinmiş $\underline{A}$ alanını ifade etmektedir.

Çarpanların toplamı olarak ifade edilen bir fonksiyonu kübik gösterilime dönüştüren M alanı aşağıdaki gibi tanımlanmıştır;

$$M\left(\left[\begin{smallmatrix} a \\ \{x\}^b \end{smallmatrix} \right]\right) = \underbrace{00, \dots, 01}_{a+1} 1, \dots, 10, \dots, 0 \quad (3.14)$$

Aşağıdaki örnekte M alanının bazı özellikleri verilmiştir

$r = \{0, 1, 2, 3\}$ ve $f = \{x_1, x_2, x_3\}$ olmak üzere;

$$M\left[\left[\begin{smallmatrix} 1 \\ \{x\}^2 \end{smallmatrix} \right]\right] = 0110$$

$$M\left\{\overline{\{x\}^2}\right\}=1001$$

$$M\left\{\overline{\{x\}^0}\right\} \bullet \left\{\overline{\{x\}^2}\right\} = M\left\{\overline{\{x\}^0}\right\} \bullet M\left\{\overline{\{x\}^2}\right\} = (0111) \bullet (1101) = 0101,$$

$$M\left\{2 \bullet \left\{\overline{\{x_1\}^1}\right\} \bullet \left\{\overline{\{x_3\}^3}\right\}\right\} = M\left\{2 \bullet \left\{\overline{\{x_1\}^1}\right\} \bullet \left\{\overline{\{x_2\}^3}\right\} \bullet \left\{\overline{\{x_3\}^3}\right\}\right\}$$

$$= 2 \circ M\left\{\overline{\{x_1\}^1}\right\} \circ M\left\{\overline{\{x_2\}^3}\right\} \circ M\left\{\overline{\{x_3\}^3}\right\} = 2(1100)(1111)(0011)$$

Burada $\circ$ birbirine bağılılığı ifade etmektedir. Görüldüğü üzere ifadede var olmayan x_2 değişkeni ifadeye bir önceki yöntemde olduğu gibi keyfi değer atanarak ilave edilmiştir.

$$M\left\{2 \bullet \left\{\overline{\{x_1\}^1}\right\} \bullet \left\{\overline{\{x_3\}^3}\right\} + 3 \bullet \left\{\overline{\{x_2\}^2}\right\}\right\} = \begin{bmatrix} 2(1100)(1111)(0011) \\ 3(1111)(0010)(1111) \end{bmatrix}$$

M 'nin karşısı R alanı ise aşağıdaki gibi tanımlanmıştır;

$\underline{A^*} = [A^{*1} \bullet A^{*2}, \dots, A^{*n}]$ kübik alan olsun. Bu durumda R alanı aşağıdaki gibi verilmektedir;

$$R(\underline{A^*}) = R(A^{*1}) + R(A^{*2}) + \dots + R(A^{*n}) \quad (3.15)$$

Eğer A^* bir küp ise $R(\underline{A^*}) = K_a \bullet R(\underline{a_1}) \bullet R(\underline{a_2}), \dots, R(\underline{a_n})$ 'dir. Burada a_i kübün i 'nci koordinatıdır. Birleşik AND Literalı $R(\underline{a_i}) = x_i^{a_i}$ aşağıdaki gibi oluşturulur;

- $c \in R$ olmak üzere bütün c 'ler için $a_{i\alpha} = a_{i\beta} = 1$ koşulunu sağlayan α ve β elemanlarını elde et. Burada $c < \alpha$ ve $c > \beta$ ve
- α ve β arasındaki sıfırlar setini aşağıdaki gibi oluştur;

$$I = \{(\alpha_1 \bullet \beta_1) \bullet \dots \bullet (\alpha_j \bullet \beta_j)\} \quad (3.16)$$

- Eğer $\alpha = 0, \beta = p-1$ ve $I \neq \emptyset$ ise $x_i^{di} = \left[\overline{\alpha^1 \{x\}^{\beta 1}} \right] \bullet \left[\overline{\alpha^2 \{x\}^{\beta 2}} \right] \bullet \dots \bullet \left[\overline{\alpha^j \{x\}^{\beta j}} \right]$ aksi halde $x_i^{di} = \left[\alpha \{x\}^\beta \right] \bullet \left[\overline{\alpha^2 \{x\}^{\beta 2}} \right] \bullet \dots \bullet \left[\overline{\alpha^j \{x\}^{\beta j}} \right]$

Örnek; $R(0111101011) = x^9 \bullet \left[\overline{5 \{x\}^5} \right] \bullet \left[\overline{7 \{x\}^7} \right]$

$$R \left(\begin{matrix} 2(0110)(1101) \\ 3(1010)(0101) \end{matrix} \right) = 2 \bullet \left[\overline{1 \{x_1\}^2} \right] \bullet \left[\overline{2 \{x_2\}^2} \right] + 3 \bullet \left[\overline{1 \{x_1\}^1} \right] \bullet \left[\overline{0 \{x_1\}^2} \right] \bullet \left[\overline{1 \{x_2\}^3} \right] \bullet \left[\overline{2 \{x_2\}^2} \right]$$

3.2.2.2. Asal bileşenlerle indirgeme yöntemi

$\underline{ON}_i^*$, $\underline{DC}_i^*$ ve $\underline{OFF}_i^*$ sırasıyla $f = i, f$, rastgele değer ve 0 küplerinin kümesi olsunlar. k 'lar için $\underline{ON}$ alanı, $k < i$ için f_i 'lerin $\underline{OFF}$ alanı olmaktadır, buradan $\underline{OFF}_i = \underline{OFF} \bigcup_{k=1}^{i-1} \underline{ON}_k$ olarak yazılabilir. m adet r tane 1'den oluşan üniversal küp $U_m = (1111, \dots, 1)(1111, \dots, 1), \dots, (11, \dots, 1)$ olarak tanımlansın. f_i 'ler için asal çarpanlar kümesi $\underline{PI}_i = U_m \# \underline{OFF}_i$ olarak elde edilebilir. Algoritma adım-adım aşağıda verilmiştir;

- Kübik notasyonda bütün i 'ler için $\underline{OFF}_i = \underline{OFF} \bigcup_{k=1}^{i-1} \underline{ON}_k$ elde et.
- Bütün $\underline{ON}_i \neq \emptyset$ 'ler için $\underline{PI}_i = U_m \# \underline{OFF}_i$ 'leri hesapla.
- $\underline{ON}$ ve $\underline{DC}$ alanları içeren ve yalnızca asal bileşenlerden oluşan $\underline{PI}^* = \bigcup_{i=1}^{p-1} \underline{PI}_i$ 'leri hesapla.
- Kübik alana ters dönüşüm R 'yi uygula ve çarpanların toplamı kanonik formunu elde et.

3.2.2.3. İndirgenemez kapsam algoritması

Eğer bir f fonksiyonunun yaklaşık minimum ifadesi elde etmek istenirse, $\underline{PI} = U_{i=1}^{p-1} \underline{PI}_i$ ile gösterilen asal çarpanlar elde edilir. $i = p-1$ 'den başlanarak $\underline{PI}_i$ alanının ilk asal bileşeni (P) alınır ve bu P tarafından kapsanan $\underline{ON}$ terimleri kesişim

işlemi ile elde edilir, $\underline{INT} = P \cap \underline{ON}_i$. Eğer $\underline{INT}$ i'nin bütün asal bileşenler tarafından kapsanırsa P'yi $\underline{PI}_i$ alanından sil, aksi halde OR işlemini uygulayarak P alanının tüm ON terimlerini kapsayan Q alanı elde edilir. Daha sonra ON terimlerine komşu olmayan keyfi değerleri AND işlemi uygulayarak elde et ve P alanından sil. İşlem bütün $p - 2, \dots, 1$ mantık seviyeli asal bileşenlere tekrarlanır.

3.3. Doğrudan Kapsama (Maskeleme) Yöntemleri

Asal bileşenlerle indirgeme yöntemleri oldukça fazla hesap zamanı gerektirmektedir. Değişken sayısı arttıkça bu yöntemler pratikliğini yitirmektedir [9]. İkili devrelerin indirgenmesinde asal bileşenler yeterli olmaktadır [4]. Ancak ÇSM fonksiyonlarının indirgenmesinde (asal ve asal olmayan) bütün bileşenlerin dikkate alınmasının gerekli olduğu gösterilmiştir [9]. Literatüre yansıyan ve bütün bileşenleri kullanarak doğrudan maskeleme yöntemini kullanan birçok yöntem vardır. Ancak bunların tamamı üç temel yönteme dayanmaktadır [10,11,12].

Her üç yöntemin de işleyişi temelde aynıdır ve genel olarak iki ana adımdan oluşmaktadır; ilkin minterm deneme-yanılma (heuristic) ile seçilir, daha sonra o minterm kapsayan çarpan oluşturulur ve çarpanın kapsadığı bütün terimler fonksiyondan silinir [12].

3.3.1. Pomper ve Armstrong algoritması

Yöntem daha önceki bölümde anlatılan kübik gösterilim üzerine kurulmuştur [10].

Literaller de uygunluk için $^i x^i \equiv x$ formunda verilmiştir ve nokta Literalı olarak adlandırılmaktadır. Yöntem üzerine aşağıdaki durumlar geçerlidir;

- f_k fonksiyonunun k maskeleri (kapsamları) elde edilsin;
- Doğruluk tablosundaki herhangi bir terim k maskesine eşitse kapsanabilir.

İndirgenemez kapsam minimum olmayabilir, ancak minimum kapsamda indirgeme yapılamaz. k terimlerden indirgenemez kapsam oluşturulmak istenirse aşağıdaki koşullar sağlanmalıdır;

- Her k terimi başka k terimleri tarafından kapsanmayan en az bir k seviyeli fonksiyon terimini kapsamalıdır.
- Hiçbir k terimi, değeri k 'dan küçük bir fonksiyon terimini kapsayamaz.
- Her k terimi oluşturulurken mümkün olduğunca fazla k seviyeli terimi fonksiyonda kapsamalıdır.

I özel kapsamı f fonksiyonunda sadece bir elemanı kapsamaktadır.

f_j sonuç kapsamı, kendi değerine eşit veya daha küçük seviyeli fonksiyon terimlerini kapsayabilir.

M ara işlem kapsamı, o ana kadar oluşturulan k kapsam kümesinde en çok fonksiyon elemanını kapsayan kapsamdır.

Geçici terim T , M ve I kapsamlarına OR işlemi uygulanarak elde edilir.

Bu tanımlar doğrultusunda algoritma aşağıda sıralanmıştır;

- r seviyeli f fonksiyonunu k seviyeli küplerinin listesini oluştur ($0 < k < r$).
- k seviyeli kapsanmayan başka küp yoksa e adımına git, aksi halde kapsanmayan küpü kapsayan I özel kapsamını tanımla. M 'i I 'ya eşitle ve c adımına git.
- Listeden kapsanmamış diğer kübü seç. Eğer listede kapsanmayan başka küp yoksa $f_j = M$ eşitle ($j = j + 1$). f_j tarafından kapsanabilir bütün küplerin kapsandığından emin ol, aksi halde kapsanmamış kübü I olarak seç. Daha sonra geçici kapsam T 'yi oluştur.
- Bütün küplerin T tarafından kapsanıp kapsanmadığını kontrol et. Eğer k 'dan küçük kapsanmayan küp varsa T 'yi iptal et ve c adımına git. Eğer bütün küpler k 'ya eşit veya ondan büyük değerli ise, M 'yi T 'ye eşit seç ve c adımına git.
- Bütün k seviyeli küpler kapsandıysa indirgenemez kapsam oluşmuştur. Aksi halde k kümesinden bir kapsam seç ve b adımına git.

3.3.2. Besslich algoritması

Yöntem ikili fonksiyonların indirgenmesinde de kullanılan genelleştirilmiş “r” seviyeli endeks sistemini kullanmaktadır [38]. Fonksiyonun endeksli doğruluk tablosu oluşturulduktan sonra, tablo gözlerindeki elemanlara r sayı tabanında değer atanır. Değer atama işlemi ikili mantık’ta da kullanılan ve doğruluk tablosunda terimler arasındaki yakınlık ölçütü olan Hamming mesafesine göre yapılmaktadır. Değer atanan elemanlardan en düşük seviyeli olanı ilk indirgenecek terim olarak seçilir ve terimi tablonun en sağ ve üst köşesine işleyen yer-değiştirme dönüşümü uygulanır. Bu dönüşümle kanonik terim elde edilir. Doğruluk tablosunda oluşturulan kanonik terimin bileşenleri oluşturulur ve indirgeme algoritması uygulanır. İndirgeme işlemi aşağıdaki işlemlere dayanmaktadır [11];

3.3.2.1. “r” seviyeli endeksleme yöntemi

“r” seviyeli endeksleme (adresleme) sistemi Karnough diyagramının genelleştirilmiş halidir. Örnek adresleme Tablo 3.3’te verilmiştir.

Tablo 3.3 Üç değişkenli endeksleme tablosu

			$X_0=1$		$X_0=1$		$X_0=1$					
				$X_0=2$		$X_0=0$		$X_0=2$				
$X_1=0$		$X_0=0$	0	1	2	11	10	9	18	19	20	} $X_3=0$
$X_1=1$			3	4	5	14	13	12	21	22	23	
$X_1=2$			6	7	8	17	16	15	24	25	26	
	}		33	34	35	44	43	42	51	52	53	} $X_3=1$
$X_1=1$			30	31	32	41	40	39	48	49	50	
$X_1=0$			27	28	29	38	37	36	45	46	47	
	}		54	55	56	65	64	63	72	73	74	} $X_3=2$
$X_1=1$			57	58	59	68	67	66	75	76	77	
			60	61	62	71	70	69	78	79	80	
			$X_2=0$			$X_2=1$			$X_2=2$			

Tablo bütün girişlere bağlı olan tüm çıkışları içermektedir. Değişkenler (girişler) 0’den r^n-1 ’e kadar adreslenmektedir. Tabloda adreslenen gözlerde $(0,1,...,r-1,d)$

fonksiyonel değerler bulunmaktadır. Tabloda herbir değişken bir sıra/sütun ile temsil edilmektedir. Kare tablo, endeksler simetrik olacak şekilde tablonun dış kenarlarından biri kendi etrafında sağa veya aşağıya doğru kopyalanarak elde edilir (Tablo 3.3'te koyu çizgiler kopyalama eksenlerini göstermektedir). Elde edilen yapı n tane uzayın 2 boyutlu eşdeğeridir. $r=2$ için tablo Karnough diyagramına dönüşmektedir. Karnough diyagramında komşu terimler arasında 1 Hamming mesafesi bulunmaktadır. Bu durum $r>2$ için genelleştirilirse; kopyalama eksenine simetrik olan endeksler 1 Genelleştirilmiş Hamming Mesafesi'ne (GHM) sahiptir. Endeksli adresleme yönteminin en büyük avantajı bir sonraki bölümde görülecek olan işaret akış diyagramının kullanımına uygun olmasıdır.

3.3.2.2. “r” seviyeli yer-değiştirme dönüşümü

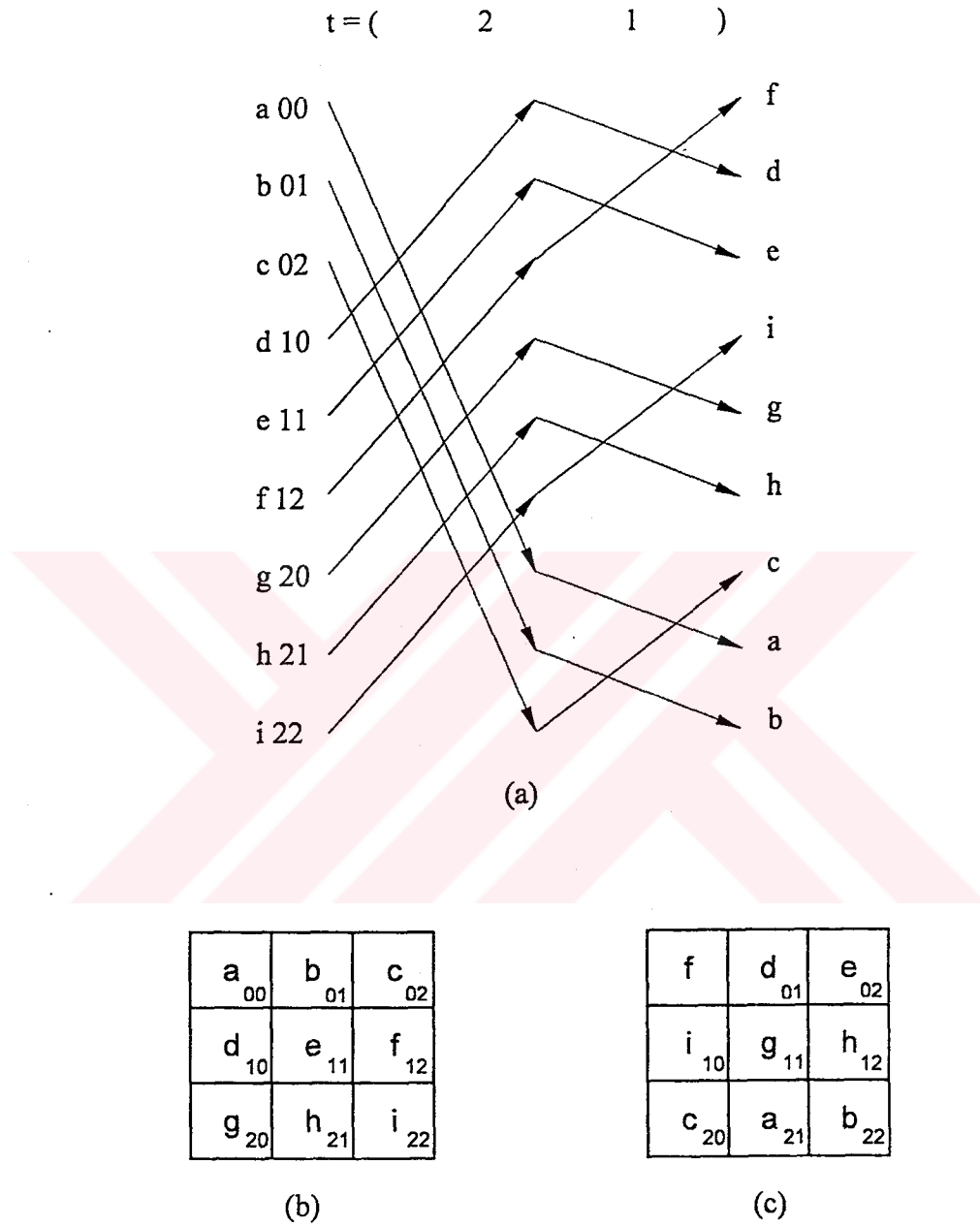
Bu dönüşüm işaret akış diyagramı yardımı ile yapılmaktadır (Şekil 3.1a). Dönüşümün en büyük özelliği ise yer-değiştiren endeksler arasında komşuluk ilişkilerinin korunması yani GHM'nin değişmemesidir. Bu dönüşüm temel olarak belirli bir endeksin 00,...,0 adresine taşınmasından ibarettir. Yer-değiştirme endeksi $\bar{i}_m$, $(i_m + \bar{i}_m) = 0$ formülünden hesaplanır. Burada $i_m, \bar{i}_m \in \{0,1,...,r-1\}$ 'dir. Örneğin şekildeki tabloda f sembolü ile verilen veri 00 endeksine (adresine) taşınısın. Bu durumda yer-değiştirme endeksi $\bar{i} = (\bar{12}) = (21)$ olarak elde edilir. Şekil 3.1b'ye yerdeğiştirme endeksinin uygulanması ile Şekil 3.1c elde edilir. Yer-değiştirme değişken sayısı kadar adımda gerçekleşmektedir.

3.3.2.3. “r” üzerinde değer atama dönüşümü (weight transform)

Dönüşüm sonucunda doğruluk tablosundaki her elemana doğal sayılar kümesinden bir sayı karşı düşürülmektedir. İşlenecek her endekse geri kalan endeksler komşulukları ile ters orantılı olacak şekilde katkıda bulunurlar. Dönüşüme öncelikle bir endekse olan GHM uzaklıklarının elde edilmesi ile başlanır. Endeks numaraları i_q ve i_r olan iki veri arasındaki mesafe “D” aşağıdaki gibi bulunur [39];

$$D = \sum_j \Delta_j(i_q, i_r) \quad (3.17)$$

$$D_{\text{maks}} = n\Delta_{\text{maks}} = n(p-1) \quad (3.18)$$



Şekil 3.1 İşaret akış diyagramı ve yerdeğiştirme örneği

İfade (3.17)'deki toplama işlemi endeks numaralarının değiştiği bütün yönlerde uygulanır. Mesafeye göre ters orantıda değişen değer katsayıları ise değer atama çarpanı yardımı ile aşağıdaki ifadeden hesaplanır;

$$w_{q,r} = \prod_{j=1}^n B^{(p-1)-\Delta_j(i_q, i_r)} \quad (3.19)$$

veya (3.17)'den

$$w(D) = B^{n(p-1)-D} \quad (3.20)$$

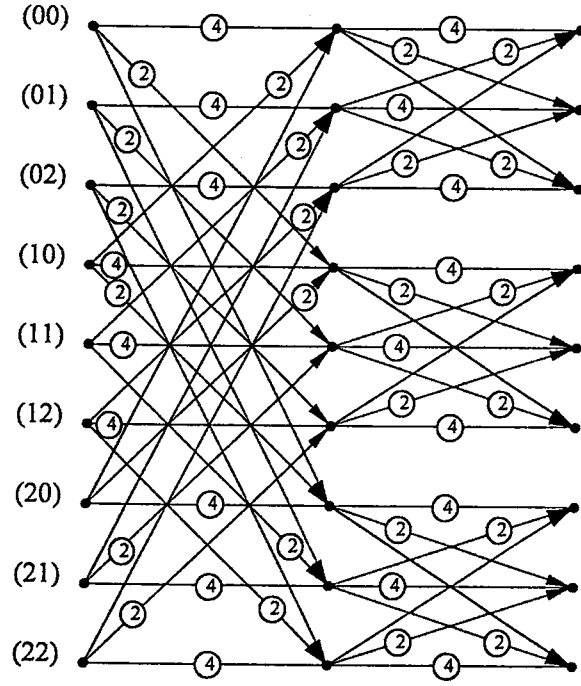
Burada B değer atama çarpanıdır ve değeri 1'den büyük seçilmelidir. Adreslere atanan değerlerin en küçük olanı kanonik terim (matrisin sol üst endeksi) olarak atanmaktadır. Aşağıdaki şekilde p=3 ve n=2 için işaret akış diyagramı verilmiştir. İşaret akış diyagramı yardımı ile GHM çarpım tablosu oluşturulmaktadır. Bu tabloyu girişlerin her durumu için ayrı-ayrı oluşturmak gerekmektedir. (22) endeksi için GHM çarpım tablosu Şekil 3.2b'de görülmektedir. Şekildeki c şıkkındaki örnek fonksiyonun elemanları oluşturulan GHM çarpım tablosunun aynı endeksli terimleri ile çarpılarak d şıkkındaki değer atama dönüşüm tablosu elde edilir.

3.3.2.4. Kontrol tablosunun oluşturulması

Adreslere atanan değerlerin sınırlarını daraltmak için fonksiyonel değerler değiştirilmektedir. Daha önce ikili fonksiyonların indirgemesinde de benzeri bir işlem yapılmıştır [39]. Bu işlemin esas amacı rastgele değerleri değer atama işleminde etkisiz kılmaktır. İkili'de $1 \rightarrow 1, d \rightarrow 0$ ve $0 \rightarrow -1$ kontrol değerleri atanmaktadır. Üç seviyeli mantık için, $2 \rightarrow 1, 1 \rightarrow 1, d \rightarrow 0$ ve $d \rightarrow -1$ atanabilir. Fonksiyonel değerlere kontrol değerleri atanarak elde edilen tabloya kontrol tablosu adı verilmektedir.

3.3.2.5. Çarpan tarama dönüşümü

Bir fonksiyonun minimal çözümünü elde etmek için asal ve asal olmayan bütün bileşenlerin oluşturulması gerektiği bilinmektedir [9]. Şekil 3.3a'da endeks numarası 00 olan terimi kapsayabilecek bileşenlerin tamamı gösterilmiştir. 00 adresini kapsayan fakat bu adresteki terimin indirgeni olmayan bileşenler de Şekil 3.3b'de verilmiştir.



(a)

$B^0=1$	$B^1=2$	$B^2=4$
$B^1=2$	$B^2=4$	$B^3=8$
$B^2=4$	$B^3=8$	$B^4=16$

(b)

0	1	0
1	2	0
1	0	1

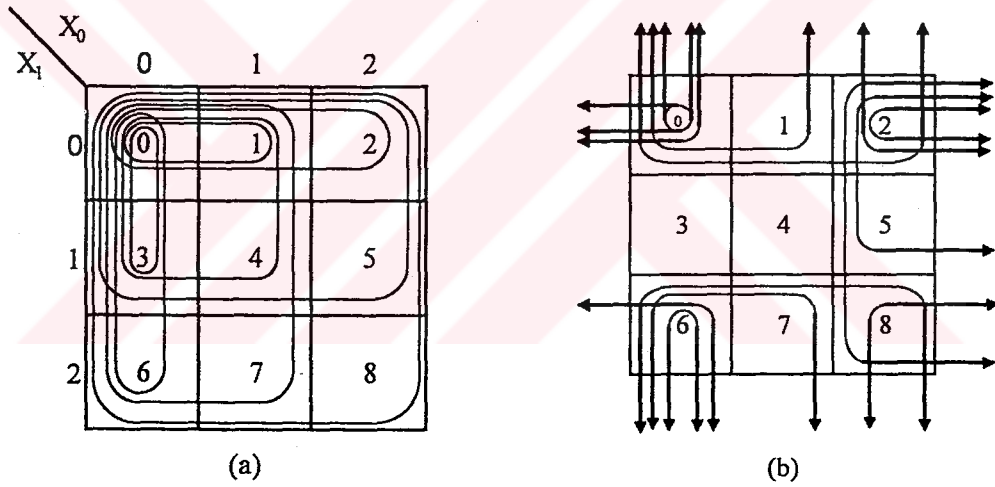
(c)

29 00	40 01	23 02
46 10	56 11	34 12
38 20	40 21	32 22

(d)

Şekil 3.2 Değer atama dönüşümü

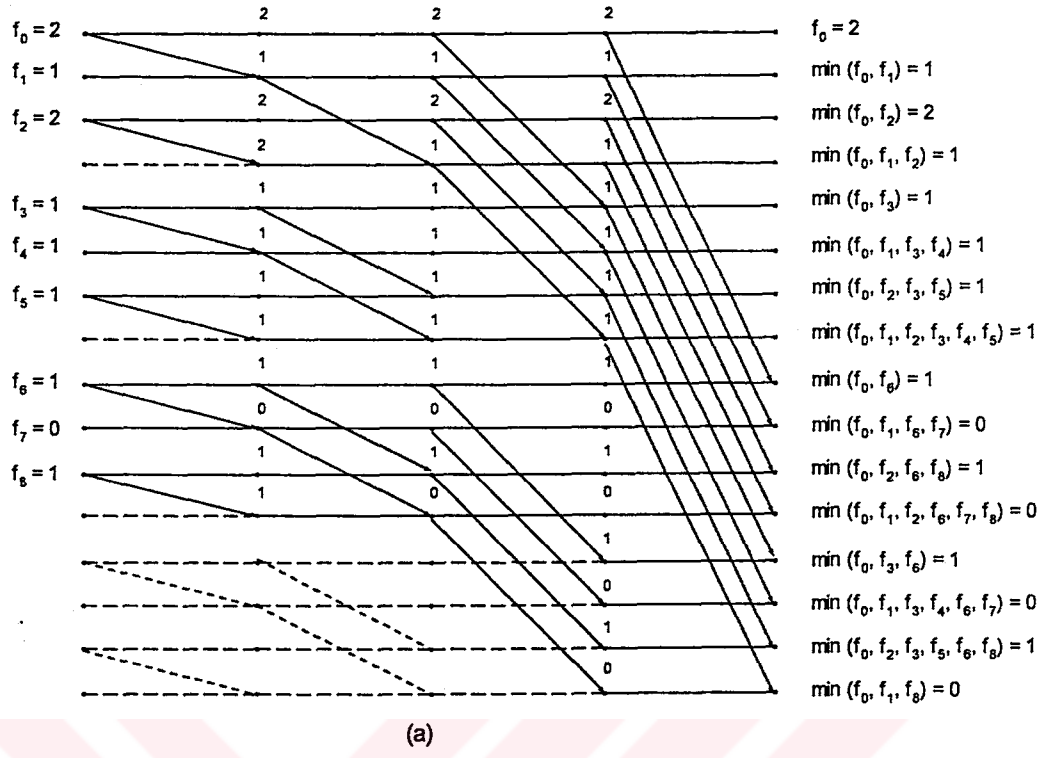
Genel olarak kanonik terimi kapsayabilen olası çarpan sayısı $2^{(r-1)n}$ tanedir, ancak bunlardan sadece r^n tanesi bileşendir. Endekslerde bulunan fonksiyonel değerlerle birlikte, olası çarpanların hangilerinin kanonik terimin gerçek çarpanı olduğunu bulmak için S dönüşümü kullanılmaktadır. Bu dönüşüm daha önce ikili devrelerin indirgenmesinde de kullanılmıştır [39]. İkili fonksiyonların indirgenmesinde kullanılan S dönüşümü ÇSM fonksiyonlarının dönüşümünde kullanılabilmesi için modifiye edilmesi gerekmektedir. ÇSM için kullanılan S dönüşümü S_r ile verilmektedir. Bu dönüşüm işaret akış diyagramı yardımı ile yapılmaktadır. Dönüşüme bütün çarpanları dahil etmek için $2^{(r-1)n} - r^n$ tane boş yer ilave etmek gerekir. Fonksiyonel değer sayısı r^n 'dir. Tablodan hareketle oluşturulan işaret akış diyagramı Şekil 3.4a'da verilmiştir. Tek bir tane sonuç elde etmek için Min işlemi uygulanmıştır. İşlem sonucu çıkan değer, o çarpanın kapsayabileceği fonksiyonel terim sayısını göstermektedir.



Şekil 3.3 00 endeksinin bileşen ve asal bileşenleri

3.3.2.6. Bileşen seçimi

Öncelikle en küçük değere sahip adres kanonik terim olarak seçilmelidir. Daha sonra yerdeğiştirme dönüşümü ile bu endeks 00,...,0 adresine taşınır ve bu adresin çarpanları oluşturulur. Bu adresteki değer eşdeğerli birden fazla çarpan tarafından kapsanıyor olabilir. Bu durumda e_i etkinlik katsayısı tanımlanır ve etkinlik katsayısı yüksek olan çarpan yaklaşık minimum bileşen olarak seçilir.



		X_0		
X_1	0	0	1	2
	1	2	1	2
	2	1	1	1

		0	1	2
0	2	0	1	2
1	1	3	4	5
2	1	6	7	8

(b)

Şekil 3.4 Üç değerli dönüşüm işaret akış diyagramı

$$e_i = \frac{m_i}{k_i} \quad (3.21)$$

Burada m_i Sr dönüşümünden elde edilmektedir. k_i ise çarpanın maliyeti olarak adlandırılmaktadır ve fiziksel olarak devreyi kurmak için kullanılan tranzistor sayısı ile orantılıdır.

Yukarıdaki işlemlere dayanan Max ve Plus (Tsum) birleştirici operatörleri için indirgeme algoritmaları aşağıdadır.

3.3.2.7. Max algoritması

- a) “r” seviyeli endeksleme ile fonksiyonu doğruluk tablosuna yerleştir.
- b) Doğruluk tablosundaki fonksiyonel değerleri kontrol değerleri ile değiştirerek kontrol tablosunu oluştur.
- c) “r” seviyeli değer atama dönüşümünü kontrol tablosuna uygula.
- d) En düşük değerli endeks 00,...,0 adresinde görülecek şekilde tabloyu yeniden organize et.
- e) S_r dönüşümünü uygulayarak kanonik terimi kapsayan bileşenleri tespit et.
- f) Etkinlik faktörünü bütün bileşenler için hesapla ve en yüksek değerli olanı yaklaşık minimum bileşen olarak seç.
- g) Yaklaşık minimum bileşeni tarafından kapsanan bütün terimleri doğruluk tablosunda rastgele (keyfi) değere dönüştür.
- h) 0’dan farklı bütün terimler kapsanınca işlemi bitir aksi halde b adımından işlemlere yeniden başlanır.

3.3.2.8. Plus (Tsum) algoritması

- a – f Max algoritması ile aynı işlemleri yürüt.
- g) Bileşenlerin kapsadığı bütün terimleri doğruluk tablosundan sil. Eğer kanonik terimin değeri silinmezden önce r-1’den farklı ise ve kapsanınca mantık 0 değerini aldıysa, kanonik terimin değerini 0 olarak ata. Aksi halde keyfi değer olarak değerlendir.
 - h) Algoritma doğruluk tablosu boşalınca, yani sadece 0’lar ve keyfi değerler kalınca sona erer. Aksi halde b adımından işleme yeniden başlanır.

3.3.3. Dueck&Miller indirgeme yöntemi

Tsum birleştirici operatörü ile ifade edilen fonksiyonlar için geliştirilen bir indirgeme algoritmasıdır. Besslich yöntemi Max operatörü için oldukça iyi sonuçlar vermektedir [40,41]. Ancak Besslich'in Tsum algoritması minimum çözümden genellikle uzak kalmaktadır. Bu sorunu aşmak için Besslich Tsum algoritması Dueck&Miller tarafından geliştirilmiştir [12]. Besslich algoritmasından farklı olarak yeni bir bileşen seçim yöntemi önerilmiştir. Diğer işlemlerde bir değişiklik olmamaktadır.

Bileşen seçiminde izolasyon faktörü (Insulation Factor-IF) kavramından yararlanılmaktadır. Bu hesaplama Besslich'in r alanı üzerinde değer atama dönüşümüne benzer yoldan yapılmaktadır. Herbir minterm için bir izolasyon faktörü hesaplanmaktadır. Hesaplama sonucunda en büyük IF değerli minterm ilk indirgenecek terim olarak seçilmektedir. Daha sonra bu mintermi kapsayabilen tüm bileşenler oluşturulur. Bu bileşenlerin herbiri için de *izafi kırılma nokta sayısı* hesaplanır (relative break count rbc). rbc sayıları, bileşen fonksiyondan silinince oluşturacağı belirsizlik noktalarının ölçütü olarak değerlendirilebilir. En düşük rbc sayısına sahip bileşen , yani silindikten sonra geri kalan fonksiyonu en basit hale getiren bileşen ilk indirgenecek çarpan olarak seçilir [40,41].

3.3.4. İndirgeme yöntemlerinin karşılaştırılması

Butler v.d. tarafından 1990 yılında Armstrong ile Dueck&Miller yöntemlerini içeren bir bilgisayar programı geliştirilmiştir. Hamlet (Heuristic Analyzer for Multiple-valued Logic Expression Translation) adlı bu program indirgenecek fonksiyona her iki indirgeme yöntemini uygulamakta ve her iki programdan daha minimum ifadeye sahip olanını “en iyi” (Gold) olarak seçmektedir. Hamlet ile yapılan analizlerde Armstrong yönteminin birçok fonksiyon için daha iyi sonuç verdiği saptanmıştır [42].

1991 yılında ise her üç algoritma ve Hamlet programı üzerine detaylı bir karşılaştırma yapılmıştır. Bilgisayar ile 2 değişkenli ve 4 seviyeli fonksiyonlar oluşturulmuş ve algoritmalar sırasıyla uygulanmıştır. Yaklaşık 7000 fonksiyon üzerinde yapılan incelemede Besslich ve Dueck&Miller algoritmalarının Armstrong

algoritmasına göre daha iyi sonuç verdiği ortaya konmuştur. Özellikle Besslich yönteminin Max operatörü ile indirgemedede belirgin olarak diğer iki yöntemden daha iyi olduğu saptanmıştır. Değişik indirgeme yöntemlerinin minterm ve bileşen seçim kriterleri topluca aşağıdaki tabloda verilmiştir [40].

Tablo 3.4 Minterm ve bileşen seçim tablosu

Algoritma	Minterm seçim kriteri	Bileşen seçim kriteri
1. Rastgele minterm/ rastgele bileşen	Rastgele	Rastgele
2. Besslich	En düşük ağırlık	En çok sıfır üreten
3. Pomper-Armstrong	Rastgele	En çok sıfır üreten
4. Dueck ve Miller	En yüksek izolasyon faktörlü	En düşük rbc
5. Hamlet (gold)	3,4 algoritmalarından en uygun olanı	

3.3.5. Maliyet tabloları ile indirgeme teknikleri

Yukarıda ayrıntılı olarak incelenen üç değişik indirgeme yöntemi, son yıllarda maliyet tabloları ile birleştirilmiş ve birçoğu bilgisayar ortamına aktarılmıştır. Bileşenler genellikle Besslich veya Dueck&Miller yöntemlerine göre seçilmektedir. Maliyet tabloları ile analiz bu aşamadan sonra devreye girmektedir. Tabloların birçoğu 4 seviyeli ve iki değişkenli mantık için Literal operatörü ile bütün Bileşenler gerçekleştirilecek şekilde hazırlanmaktadır. Bileşen seçildikten sonra, bu çarpanı en ucuz şekilde gerçekleyen Literal operatörleri tablodan seçilmektedir. Bu konuda örnek programlar/algoritmalar için [41,43,44]'e bakılabilir.

4. ÇSM CMOS AKIM MODLU DEVRE TASARIMI

Son yıllarda ÇSM CMOS akım modlu devrelere (Current Mode CMOS Logic CMCL) ilgi giderek artmaktadır. CMOS akım modlu devrelerin ilgi odağı olmasının üç temel sebebi vardır; birincisi, CMOS akım modlu devrelerin üretim işlemi ikili mantık CMOS üretim prosesi ile aynıdır. İkincisi, CMOS akım modlu devrelerin güçlü mantık fonksiyonelliğidir. Yani birçok mantık işlemi ÇSM akım modlu devrelerle kolayca gerçekleştirilebilmektedir [43,45]. Üçüncüsü, akım modlu devrelerin düşük besleme gerilimlerinde sorunsuz olarak çalışmasıdır [46].

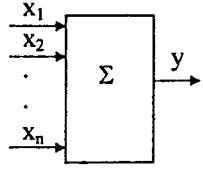
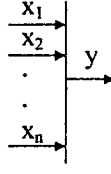
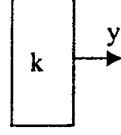
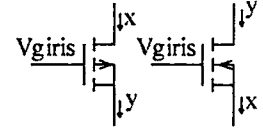
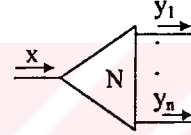
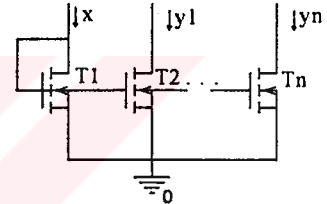
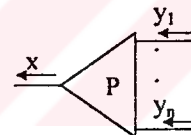
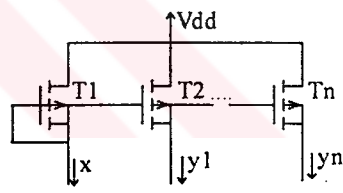
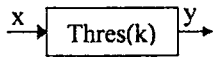
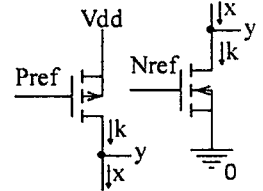
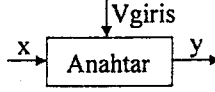
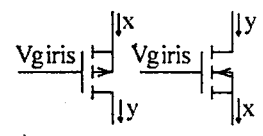
Ancak bu üstün yanlarına rağmen ÇSM CMOS akım modlu devrelerin oldukça ciddi sorunları bulunmaktadır. CMCL devreler ikili akım modlu devrelere kıyasla daha yavaş çalışmaktadır. Bunun nedeni ise ÇSM devrelerin çalışma akımının ikili devrelere göre çok daha küçük olmasıdır [47]. Ayrıca ÇSM CMOS akım modlu devreler ikili devrelere göre çok daha fazla statik güç tüketmektedir [46]. Bu sorunu aşmak için dinamik devreler geliştirilmiştir. Ancak bu devreler normal devrelere göre ilave anahtarlanmış transistor çifti ve saat işareti kullanırlar [22]. Anahtarlanmış transistor çifti kullanan devre örnekleri için [21,22,48]'e bakılabilir.

CMOS akım modlu devre tasarımında iki değişik tasarım yöntemi kullanılmaktadır. İlk tasarım örnekleri hem akım modlu hem gerilim modlu devrelerden oluşan hibrit devrelerdir. Diğer tasarım yaklaşımı ise tamamen akım modlu devreler üzerine kuruludur. Her iki tasarım yöntemi aşağıda ayrıntılı olarak incelenecektir.

4.1. Temel Devreler

Temel devreler, CMOS akım modlu devrelerin tasarımında, kontrol işaretinin oluşturulması, akımın kopyalanması, işaretin yönünün değiştirilmesi vb. işlemlerde kullanılmaktadır. Birçok devre tasarımında kullanılmış olan temel devrelerin yerine getirdiği işlemler ve devre gerçeklemeleri aşağıdaki tabloda topluca verilmiştir [43,45,49].

Tablo 4.1 Temel devreler

Yapı bloğu adı	Mantık işlemi	Sembol	Devre gerçeğemesi
Sum	$y=x_1+x_2+\dots+x_n$		
Sabit (constant)	$y=k$		
NMOS ve PMOS akım aynaları	$y_i=a_i x$ $i=1,2,\dots,n$ (a_i =ayar faktörü)		
			
P ve N tipi Threshold	<u>N tipi</u> ($x \geq k$) ise y =ikili yüksek aksi halde y = ikili düşük		
P ve N tipi anahtar	Anahtar açık ise $y=x$ Aksi halde $y=0$		

Bu operatörlerin devrede kullanılış amaçları ve çalışma prensipleri aşağıda ayrıntılı olarak verilmiştir. Tasarım için UMC018 transistor parametrelerinden yararlanılmıştır. Transistorların SPICE parametreleri EK A'da verilmiştir. UMC018 teknolojisi ile gerçekleştirilebilen en küçük boyutlu transistor $W_{min}=0.24\mu m$, $L_{min}=0.18\mu m$ 'dir. I_0 akımı mantık bir seviyesine karşı düşürülmüş ve $10\mu A$ olarak seçilmiştir. Bu teknolojiye maksimum besleme gerilimi $1.8V$ 'tur.

4.1.1. Akımların lineer toplanması (Sum)

Sum devresinin temel işlemi akımları lineer toplamaktır. Tasarımı için herhangi bir aktif veya pasif elemana gerek yoktur. Sum temel devreler içerisinde birden çok işareti birleştiren tek elemandır [43,45]. Sum devresinden daha sonra görüleceği üzere Tsum operatörünün tasarımında yararlanılacaktır.

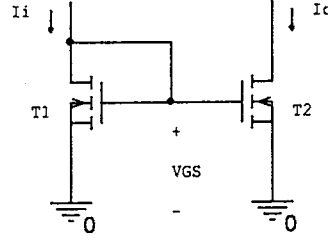
4.1.2. Sabit akım kaynağı (Constant)

Akımın yönüne bağlı olarak iki tip sabit akım üretici vardır. Bu devre ile herhangi bir mantık seviyesi oluşturulabilir. Örneğin UMC018 parametrelili transistorlar ile $I_0=10\mu A$ mantık 1 seviyesini oluşturmak için $W/L=1$ transistorların girişine $P_{ref}=800mV$ ve $N_{ref}=594mV$ uygulamak yeterli olacaktır. Herhangi bir mantık seviyesi transistor boyutları değiştirilerek kolayca elde edilebilir [43,49,50].

4.1.3. NMOS ve PMOS akım aynaları

CMOS ÇSM akım modlu devrelerin tasarımında en çok kullanılan devre akım aynalarıdır. Birçok uygulama için basit akım aynası yeterli olmaktadır. Akım aynası genel olarak akımların yönünün değiştirilmesi, tek olan çıkış sayısının artırılması (fanout), akımların kopyalanması ve akımların transistor boyutları ile orantılı bir katsayı ile çarpılması ile değişik mantık seviyelerinin elde edilmesinde kullanılmaktadır [43,46,49,50,51].

Şekil 4.1'de verilen basit NMOS akım aynasının akım gerilim bağıntıları transistorların doymada çalıştıkları varsayımı ile (4.1) ifadesindeki gibi verilebilir [52];



Şekil 4.1 Basit CMOS akım aynası

$$I_i = \frac{\mu C_{ox}}{2} \left(\frac{W}{L} \right)_1 [V_{GS} - V_T]^2 \quad (4.1)$$

$$I_o = \frac{\mu C_{ox}}{2} \left(\frac{W}{L} \right)_2 [V_{GS} - V_T]^2 \quad (4.2)$$

Transistorların geometri dışında eş özellikte olduğu göz önüne alınırsa giriş ve çıkış akımlarının oranı;

$$\frac{I_o}{I_i} = \frac{(W/L)_2}{(W/L)_1} \quad (4.3)$$

şeklindedir ve tam olarak 1 yapılabilir.

ÇSM devrelerde en çok kullanılan yapı bloğu olan basit akım aynası için detaylı gecikme ve hata analizi aşağıda verilmiştir.

Devrenin küçük işaret analizi yapılırsa giriş ve çıkış arasında C_{gs}/g_m ile orantılı bir gecikmenin olduğu görülebilir. Devrenin transfer fonksiyonu aşağıdaki gibidir;

$$\frac{I_o(s)}{I_i(0)} = \frac{g_{m1}}{g_{m2}} \frac{1}{1 + s(C_{gs1} + C_{gs2})/g_{m1}} \quad (4.4)$$

Elde edilen tek kutuplu transfer fonksiyonundan zaman sabiti;

$$\tau = \frac{(C_{gs1} + C_{gs2})}{g_{m1}} \quad (4.5)$$

olarak elde edilir. Tek kutuplu fonksiyonun darbe işaret cevabı aşağıdaki gibi verilmektedir;

$$I_o(t) = \frac{g_{m2}}{g_{m1}} i_i (1 - e^{-t/\tau}) \quad (4.6)$$

Bu ifadeden hareketle, çıkış akımının yarı değerine yükselene kadar geçen süre (propagasyon gecikmesi) aşağıdaki ifadeden hesaplanabilir;

$$t_p = 0,69\tau = 0,69 \left(\frac{C_{gs1} + C_{gs2}}{g_{m1}} \right) \quad (4.7)$$

Propagasyon gecikmesi, akım seviyeleri (g_m) artırılarak azaltılabilir. Ancak bu durumda transistor boyutları dolayısıyla parazitik kapasiteler de artacaktır.

Akım modlu devreler eşik gerilimi ve diğer değişimlerden dolayı kötü gürültü sınırlarına sahiptir. Transistordan akan akım $(V_{gs} - V_t)^2$ ile orantılı olduğundan eşik gerilimi V_t 'de oluşacak değişimler elemandan akan akımı etkileyecektir. Bu etkinin yanı sıra darbe genişlik modülasyonu dolayısıyla geometri ve akım hataları da çıkış akımı üzerine etkilidir. Akım üzerine olumsuz etkiye sahip her üç etken aşağıdaki gibi modellenabilir;

$$\frac{\Delta I}{I} = \frac{\Delta(W/L)}{W/L} + \frac{2\Delta V_t}{V_{gs} - V_t} + \lambda \Delta V_{ds} \quad (4.8)$$

Özellikle dar kanallı transistörlerde akım aynasının hatası daha büyük olmaktadır. Zira λ kanal genişlik modülasyon parametresi L ile ters orantılıdır.

4.1.4. Gerilim çıkışlı akım karşılaştırıcı (Thershold) devresi

Thershold devresi akımı gerilime çevirmek için kullanılır. Tabloda da görüldüğü üzere N ve P tipi olmak üzere iki çeşit devre mevcuttur. N tipi devre P tipi devrenin eşleniğidir. Thershold devreleri daha önce ikili akım modlu devrelerin tasarımında da kullanılmıştır. ÇSM akım modlu hibrit tipi operatör tasarımında görüleceği üzere thershold devresi ikili kontrol gerilimlerini oluşturmak için kullanılmaktadır. Bu

devrede k transistordan akabilecek maksimum akımı göstermektedir. Daha sonra görüleceği üzere bu devreden hibrit Döngü ve Literal operatör devrelerinin gerçekleştirilmesinde yararlanılacaktır [49]. Thershold devreleri ile ilgili detaylı çalışmalar için [47,53]'e bakılabilir. Threshold devreleri akımı gerilime çevirmek için kullanılır ve genel ifadesi aşağıdaki gibidir [27];

$$\text{Thres}(x,k)=\begin{cases} \text{ikili yüksek } x \geq k \text{ ise} \\ \text{ikili dusuk aksi halde} \end{cases} \quad (4.9)$$

Burada $k \in r$ 'dir.

4.1.5. Basit anahtar (Switch)

Anahtar kapısından gerilim ile kontrol edilen P tipi veya N tipi transistordan oluşmaktadır. Anahtar açık iken (N tipi anahtar için $V_{\text{giriş}}$ ikili yüksek ise) devreden akım akmaktadır. Anahtar elemanları ÇSM hibrit tipi devrelerin genellikle çıkış katında yer almaktadır ve kontrol katı tarafından kontrol edilerek çıkışta ÇSM akımlarının yeniden oluşturulmasında kullanılmaktadır [45,49].

4.2. ÇSM Operatörlerinin Devre Gerçeklemeleri

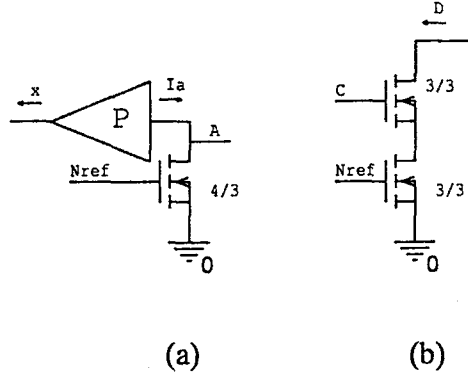
Günümüzde ÇSM operatörlerinin tasarımı için iki değişik tasarım tekniği kullanılmaktadır. Özellikle doksanlı yıllara kadar hibrit modlu devre tasarım tekniği ön planda olmuştur. Son yıllarda CMOS devrelere artan ilgi dolayısıyla tamamen akım modlu devre tasarım tekniği önem kazanmıştır. Her iki tasarım yöntemi aşağıda ayrıntılı olarak verilmiştir.

4.2.1. Hibrit CMOS ÇSM devreleri

ÇSM akım modlu devrelerin ilk tasarımlarında I^2L ve ECL teknolojisi ön plandadır. Bu devreler genellikle ikili devrelerin ÇSM mantığına genişletilmesi ile elde edilmiştir. ÇSM CMOS akım modlu devrelerin giriş katından sonra gerilim modlu çalışan bir kontrol katı ilave edilmektedir. Bu kontrol katı ile çıkışta istenen ÇSM akım işaretleri yeniden oluşturulmaktadır [1].

Şekildeki yapı hem CMCL hem de VMCL (Voltage Mode CMOS Logic) devrelerinden oluşmaktadır ve hibrit tasarım yöntemi olarak bilinmektedir. Yapı üç ana bölümden oluşmaktadır; giriş, kontrol ve çıkış. Giriş ve çıkış blokları ÇSM akım modlu çalışan devrelerden, kontrol kısmı ise gerilim modlu ikili devrelerden oluşmaktadır. Giriş genellikle akım aynaları ve Thershold devre elemanlarından oluşur. Çıkış bloğu da anahtarlar, sabit akım kaynakları (constant) ve akım aynalarından oluşur. Giriş ve çıkış devreleri için birer örnek Şekil 4.3'te verilmiştir. Şekil 4.3'deki giriş devre örneğinde, x giriş işareti P tipi akım aynası ile yön değiştirmekte ve kapısına referans gerilimi uygulanmış olan Threshold devresinin kaynağına verilmektedir. Bu devre giriş akımının eşik akımından büyük olması durumunda A düğümünde kontrol katında kullanılacak olan ikili yüksek gerilim işareti oluşturur. Şekil 4.3(b)deki devre ise c düğümünden gerilim ile kontrol edilen basit anahtar ve referans akım oluşturma transistöründen ibarettir. c düğümüne Şekil

4.2'deki kontrol katının gerilimi uygulanmakta ve Nref ile elde edilen D akımı kontrol edilmektedir.



Şekil 4.3 Giriş (a) ve çıkış (b) devre örnekleri

Giriş katının görevi ÇSM giriş akımı değerlerine karşı thershold elemanını kullanarak ikili gerilim değerlerini oluşturmaktır. Kontrol katında, evirici, NAND, NOR veya karmaşık CMOS kapılar ile giriş katından alınan gerilim değerlerine göre kontrol işaretleri üretilir. Bu kontrol işaretleri çıkıştaki anahtarları kontrol etmek için kullanılır. Hibrit modlu devrelerde kullanılan ikili mantık devrelerinin simülasyonu için, transistor besleme gerilimi UMC018 teknolojisinde 1,8V'u aşmadığından, mantık devrelerinde 5V kullanan SPICE standart kütüphanesi EVAL.LIB'den yararlanılamamaktadır. Evirici, AND ve OR kapıları simulasyon için kullanılan tranzsitorlarla tasarlanmıştır. Temel CMOS ikili devre tasarımı için [54]'e bakılabilir.

4.2.1.1. Min operatörünün devre gerçekleştirilmesi

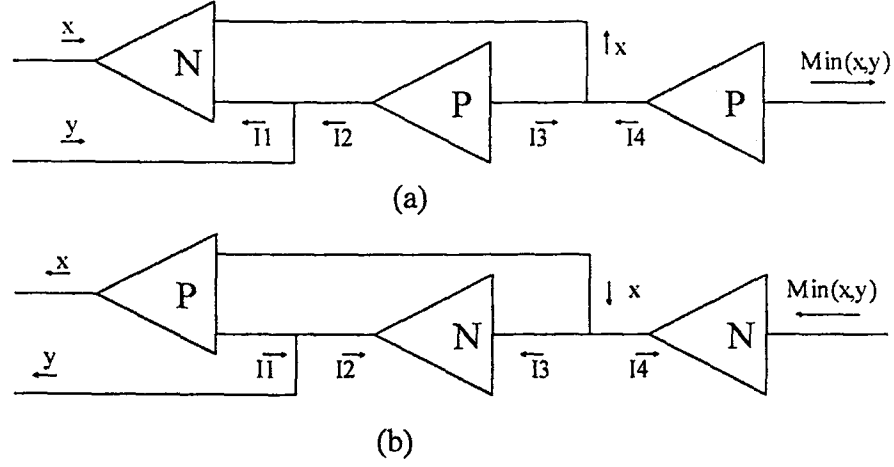
Min operatörünün tasarımı sınırlı fark (Truncated difference-Tdif) operatörüne dayanmaktadır. Bu operatörün genel ifadesi (2.7)'de verilmiştir [32,55].

Min operatörünün tanımı da Tdif operatörüne bağlı olarak aşağıdaki gibi verilebilir;

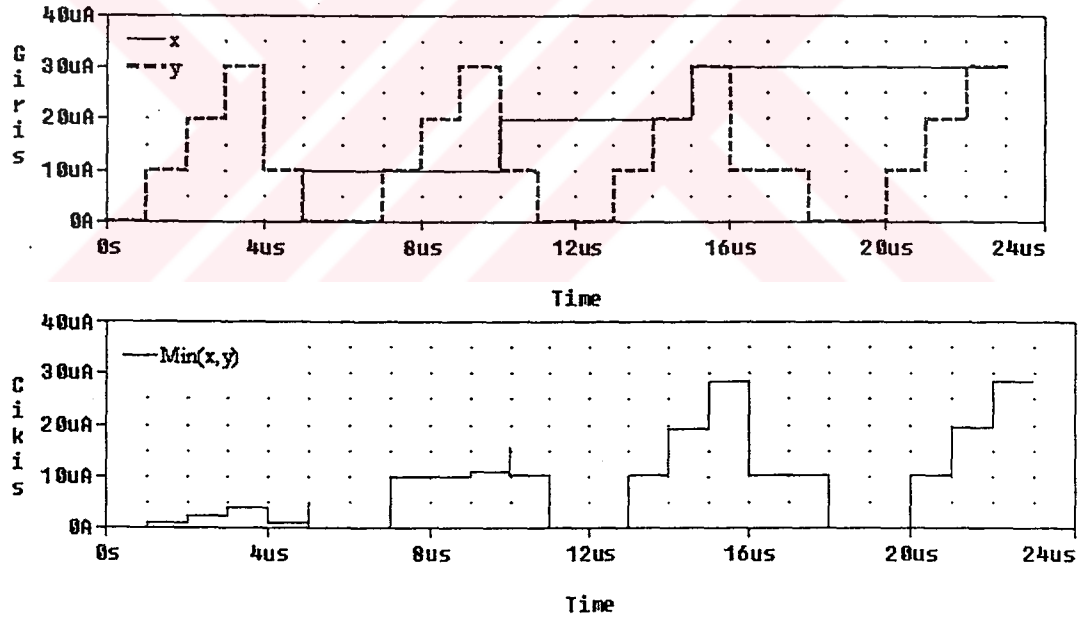
$$\text{Min}(x,y)=y-(x\ominus y)=y\ominus(y\ominus x) \quad (4.10)$$

Çıkarma işlemi, çıkarılacak akımın yönünün akım aynaları ile değiştirilmesi ve çıkartılana eklenmesi ile gerçekleştirilir. Min operatörü temel olarak iki giriş akımından küçük olanını çıkışa yansıtmaktadır. Min operatörünün, giriş akımının yönüne göre

iki değişik devre tipi bulunmaktadır. Devreler ve simülasyon sonuçları Şekil 4.4 ve 4.5'te verilmiştir.



Şekil 4.4 Hibrit Min operatörü devresi



Şekil 4.5 Hibrit Min operatörü simülasyon sonuçları

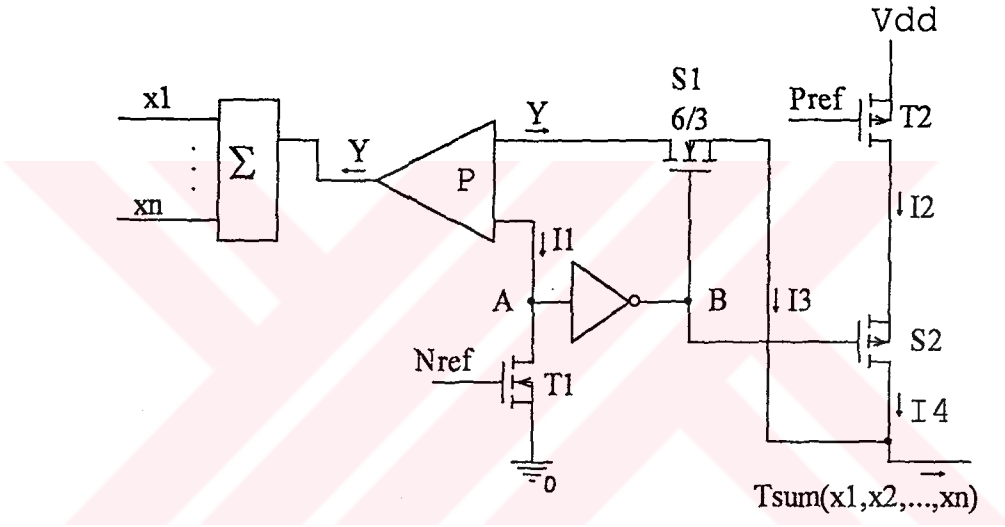
Bu devrelerde $I_3 = (x-y)I_0$ ve $I_4 = xI_0 - I_3$ 'tür. Şekildeki devrenin çalışma prensibini anlamak için şu iki durumun göz önüne alınması gerekir; $x < y$ ve $x \geq y$.

Durum 1; ($x < y$), $I_1 = xI_0$ ve $I_2 = 0$, dolayısıyla $I_3 = 0$ ve $I_4 = xI_0 - I_3 = xI_0$ olacaktır. Buradan da $\text{Min}(x,y) = x$ olduğu anlaşılır.

Durum 2; ($x \geq y$), $I1 = xI_0$ ve $I2 = xI_0 - yI_0$ dolayısıyla $I3 = (x - y)I_0$ ve $I4 = xI_0 - I3 = xI_0 - (x - y)I_0 = yI_0$ olacaktır. Bu durum da $\text{Min}(x, y) = y$ anlamına gelir.

4.2.1.2. Tsum operatörünün (sınırlı toplam) devre gerçekleştirilmesi

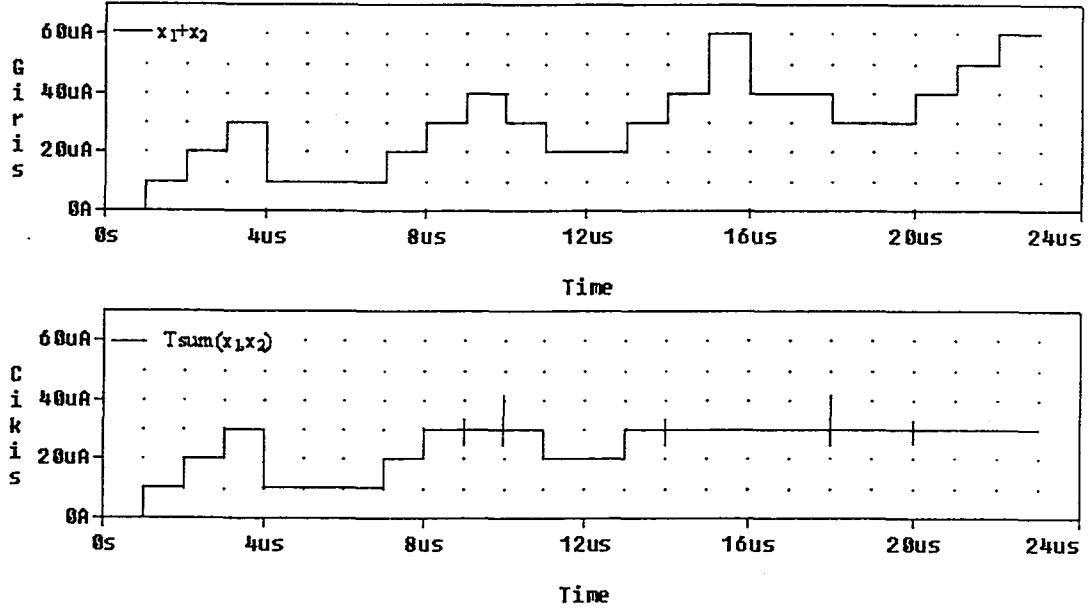
Tsum operatörünün görevi çıkış akımını $(r-1)I_0$ ile sınırlamaktır. Bu operatör, n değişkenli r seviyeli bir fonksiyon için $\text{Min}(x_1 + x_2 + \dots + x_n, r-1)$ olarak Min fonksiyonu ile gerçekleştirilebilir. Fakat bu gerçekleştirme yöntemi 3 seviye akım aynalaması ve çıkarma işlemi gerektirdiğinden çıkış işareti bozulabilmektedir. Bu sorunu aşmak üzere aşağıdaki şekildeki devre yapısı kullanılabilir [49];



Şekil 4.6 Hibrit Tsum operatörü devresi

Bu devre Şekil 4.2'deki yapıya göre tasarlanmıştır. Bu devrede T1 transistörünün boyutu $r-2$ den büyük mantık seviyelerini taramak için $W1/L1 = (r-1-0.5)$ olarak seçilmiştir. T2 transistörü da S1 anahtarı açık iken $I2 = (r-1)I_0$ akımını oluşturmak üzere $W2/L2 = (r-1)$ olarak boyutlandırılmıştır. Burada da devre çalışmasını anlamak için, $0 \leq Y < (r-1)$ ve $(r-1) \leq Y$ durumlarını inceleyelim.

Durum 1; ($0 \leq Y < (r-1)$), $I1 = YI_0$ ve A düğümündeki gerilim ikili düşük olacaktır. B düğümündeki gerilim ise ikili yüksek seviyede olacaktır. S1 anahtarı açık ve S2 kapalı olacaktır. $I3 = YI_0$ ve $I4 = 0$ olacağından çıkış akımı $I5 = I3 + I4 = YI_0$ olacaktır. Yani çıkış mantık seviyeleri Y tarafından belirlenecektir ve çıkış akımı $(x_1 + x_2 + \dots + x_n)$ olacaktır.

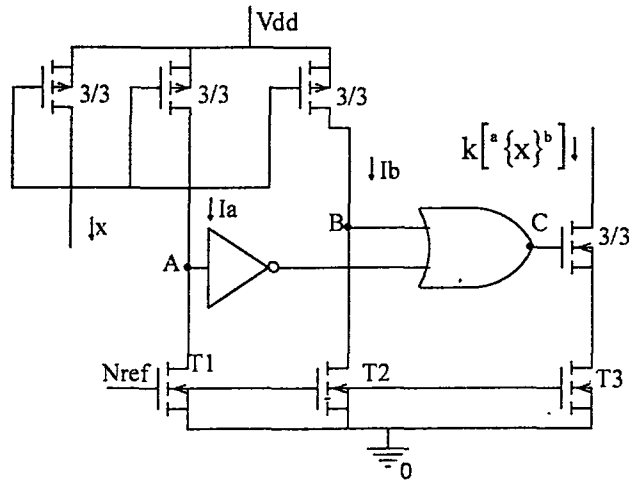


Şekil 4.7 Hibrit Tsum operatörü simülasyon sonuçları

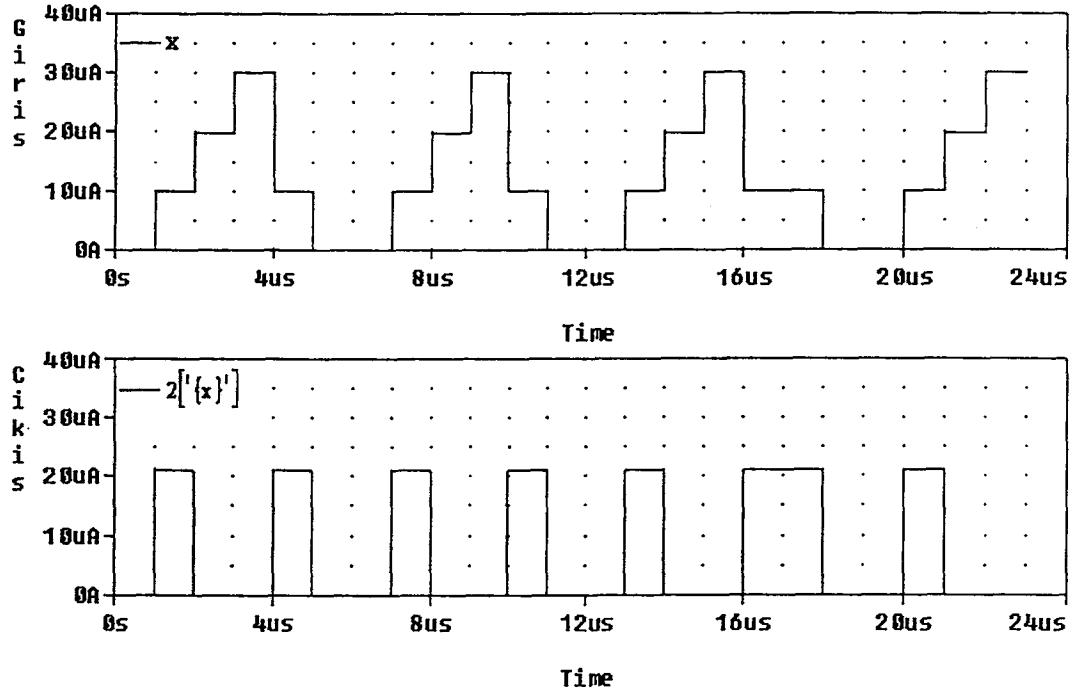
Durum 2; $(r-1) \leq Y$ ise, $I_1 = (r-1-0.5)I_0$ ve A düğümündeki gerilim ikili yüksek olacaktır. B düğümündeki gerilim ise ikili düşük olacaktır. Yani S1 kapalı ve S2 açık olacaktır. $I_3 = 0$ ve $I_4 = (r-1)I_0$ dolayısıyla çıkış akımı da $(r-1)I_0$ ile sınırlandırılmış olacaktır.

4.2.1.3. Literal operatörünün devre gerçekleştirilmesi

Literal operatörünü gerçeklemek için kullanılabilecek devre örneği Şekil 4.8'de verilmiştir [49].



Şekil 4.8 Hibrit Literal operatörü devresi



Şekil 4.9 Hibrit Literal operatörü simülasyon sonuçları

Bu devrede T3 transistörünün boyutu Literalin k çarpanını oluşturmak için $W3/L3=k$ olmak zorundadır. Eğer $k=r-1$, yani herhangi bir fonksiyonel değere eşit ise iç Literal $[^a\{x\}^b]$ gerçekleşmiş olacaktır. Şekildeki devrede T1, $(a-0,5)I_0$ akım seviyeleri için threshold ve T2 de $(b+0,5)I_0$ akım seviyeleri için threshold işlemini yerine getirmektedir. Devrenin çalışmasını incelemek için şu üç durum göz önüne alınsın; $0 < x < a$, $a \leq x \leq b$ ve $b < x < r$.

Durum1; Eğer $0 < x < a$ ise, $I_a = I_b = xI_0$. A ve B düğümlerindeki gerilimler ikili düşük seviyede olacaktır. Aynı zamanda C düğümündeki gerilim seviyesi de düşük olduğundan, C düğüm gerilimi ile kontrol edilen anahtar kapalı olacaktır. Çıkış akımı sıfır olduğundan $k[^a\{x\}^b] = 0$ olacaktır.

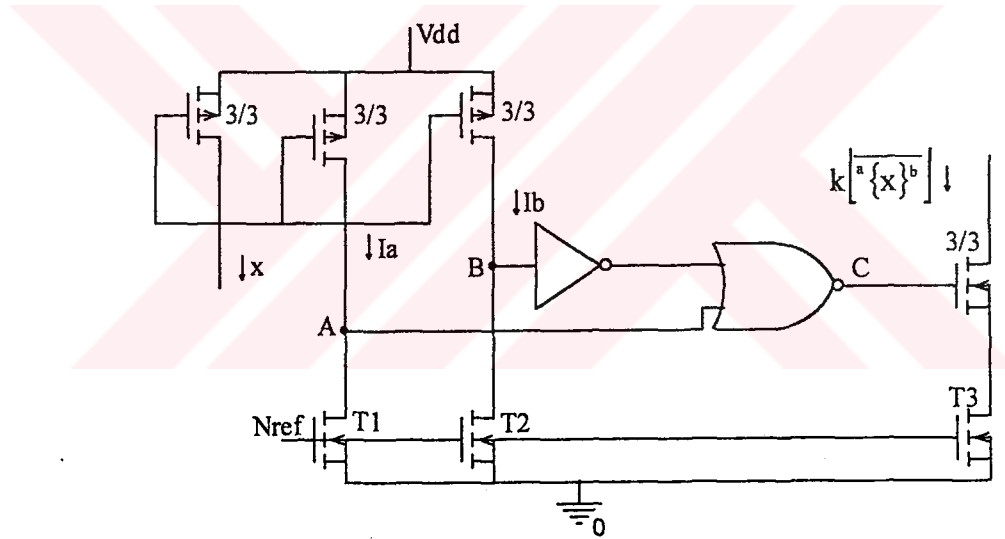
Durum 2; Eğer $a \leq x \leq b$ ise, $I_a = (a-0,5)I_0$ ve $I_b = xI_0$ olacaktır. A ve B düğümlerindeki gerilim seviyeleri de sırasıyla ikili yüksek ve düşük olacaktır. Dolayısıyla C düğümündeki gerilim seviyesi yüksek olacaktır ve C düğümü tarafından kontrol edilen anahtar açık olacaktır. Çıkış akımı da T3 transistörünün boyutuna bağlı olacaktır. Örneğin T3'ün boyutu k ile orantılı ise çıkış akımı kI_0 olacaktır. Yani $k[^a\{x\}^b] = k$ olacaktır.

Durum 3; Eğer $b < x < r$ ise, $I_a = (a-0,5)I_0$ ve $I_b = (b+0,5)I_0$ olacaktır. A ve B düğümlerindeki gerilim seviyelerinin ikisi de ikili yüksek seviyede olacaktır. C düğümü düşük seviyede olacak ve anahtar da kapalı olacağından çıkış akımı sıfır olacaktır. Yani $k\left[\begin{smallmatrix} a \\ x \end{smallmatrix}\right]^b = 0$ olacaktır.

T3 transistörünün boyutu değiştirilerek değişik k (çarpım faktörü) mantık seviyeleri elde edilebilir. Örneğin $2\left[\begin{smallmatrix} 1 \\ x \end{smallmatrix}\right]^2$ oluşturmak için $W1/L1=1/2$, $W2/L2=5/2$ ve $W3/L3=2/1$ olmalıdır.

4.2.1.4. Eşlenik Literal operatörü gerçeeklemesi

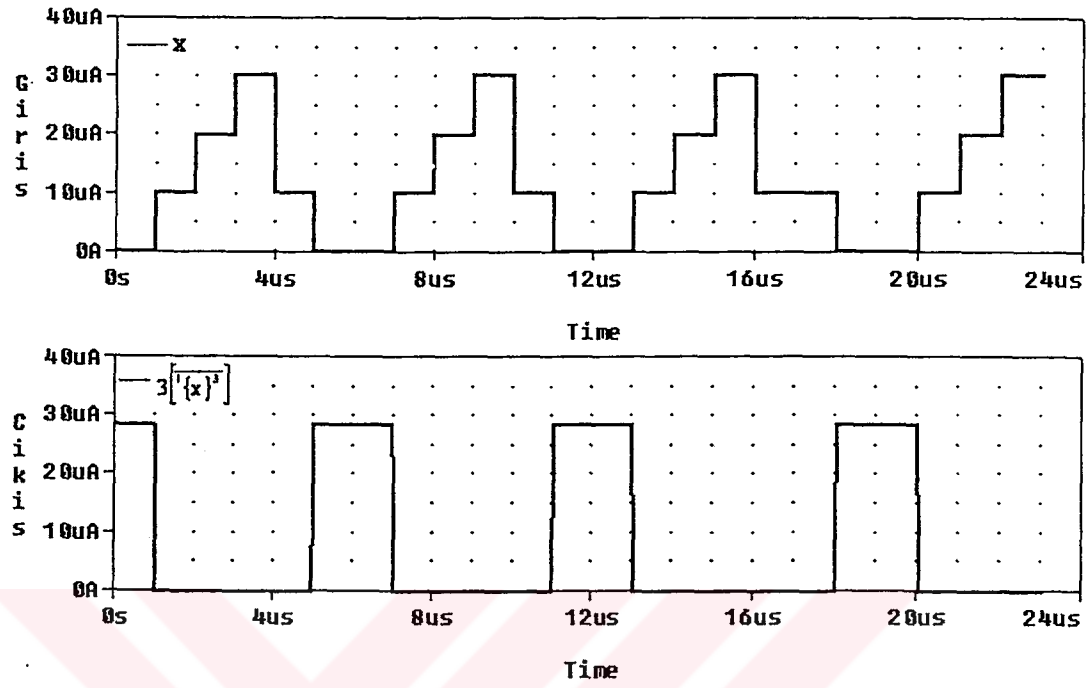
Yukarıdaki Şekil 4.8 eşlenik Literal gerçeeklemesi için geliştirilebilir önerilen devre aşağıdaki gibidir [49];



Şekil 4.10 Hibrit eşlenik Literal operatörü devresi

Eşlenik Literal fonksiyonu matematik ifadesi (2.3)'te verilmiştir. Şekil 4.8 ve 4.10'daki devreler birbirine çok benzemektedir. Sadece C düğümünde Şekil 4.8'deki devrenin eşleniğini almak için eşlenik fonksiyonu gerçeekleyen elemanlar ilave edilmiştir. Daha evvelki Literal devresine bakılırsa eğer $a=0$ olursa $k\left[\begin{smallmatrix} a \\ x \end{smallmatrix}\right]^b$ Literalinin gerçeeklenemez olduğu görülür. Ancak bu Literal eşlenik olarak yeni devre ile $k\left[\begin{smallmatrix} b+1 \\ x \end{smallmatrix}\right]^{r-1}$ gerçeeklenebilir. Örneğin $k\left[\begin{smallmatrix} 0 \\ x \end{smallmatrix}\right]^1$, $k\left[\begin{smallmatrix} 2 \\ x \end{smallmatrix}\right]^3$ olarak gerçeeklenebilir.

(4'lü sayı tabanı kullanan sistem için). Tablo 4.2'de 4 seviyeli mantık için mevcut olan Literallerin tasarımında transistorların boyutları topluca verilmiştir.



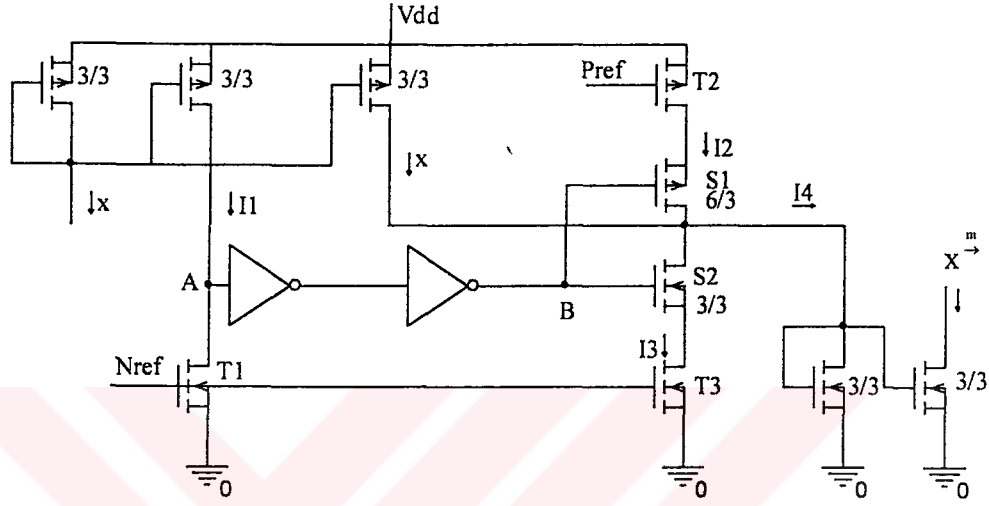
Şekil 4.11 Hibrit eşlenik Literal operatörü simülasyon sonuçları

Tablo 4.2 Literal operatörleri için transistor boyutları

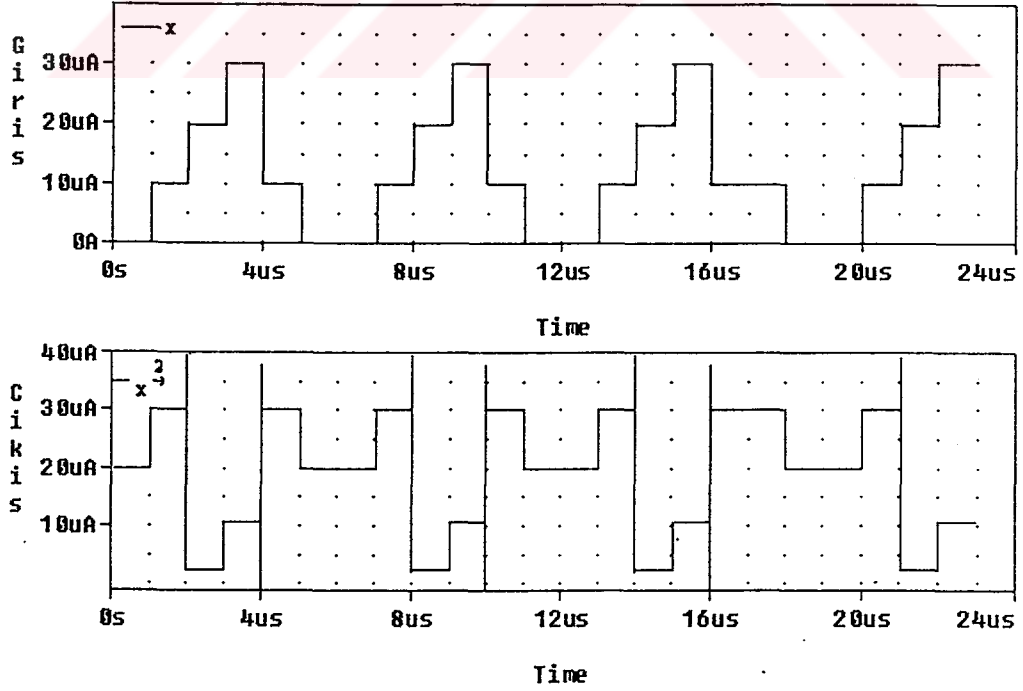
Literal operatörü	Devre gerçeklemesi	Transistor boyutları (W/L)		
		T1	T2	T3
$k[{}^0\{x\}^0]$	$k[{}^1\{x\}^3]$	1/2	7/2	k/1
$k[{}^1\{x\}^1]$	$k[{}^1\{x\}^1]$	1/2	3/2	k/1
$k[{}^2\{x\}^2]$	$k[{}^2\{x\}^2]$	3/2	5/2	k/1
$k[{}^3\{x\}^3]$	$k[{}^3\{x\}^3]$	5/2	7/2	k/1
$k[{}^0\{x\}^1]$	$k[{}^2\{x\}^3]$	3/2	7/2	k/1
$k[{}^1\{x\}^2]$	$k[{}^1\{x\}^2]$	1/2	5/2	k/1
$k[{}^2\{x\}^3]$	$k[{}^2\{x\}^3]$	3/2	7/2	k/1
$k[{}^0\{x\}^2]$	$k[{}^3\{x\}^3]$	5/2	7/2	k/1
$k[{}^1\{x\}^3]$	$k[{}^1\{x\}^3]$	1/2	7/2	k/1
$k[{}^0\{x\}^3]$	Sabit k	--	--	--

4.2.1.5. Döngü operatörü

Döngü operatörleri daha evvel de bahsedildiği üzere; Saat yönü ve ters saat yönü Döngü operatörü olmak üzere iki tanedir. Sadece saat yönü Döngü operatörünü göz önüne alalım. Döngü operatörünün örnek bir gerçeklemesi Şekil 4.12’de verilmiştir [49]. Simülasyon sonuçları da Şekil 4.13’te verilmiştir.



Şekil 4.12 Hibrit Döngü operatörü devresi



Şekil 4.13 Hibrit Döngü operatörü simülasyon sonuçları

Bu devrede T1 $(m-0.5)I_0$ akım değerleri için Thershold elemanı olarak görev yapmaktadır. T2 ve T3'ün boyutları sırasıyla $W2/L2=r-m$ ve $W3/L3=m$ olarak seçilmiştir. B düğümünde anahtar elemanlarının uygun şekilde çalışabilmesi için ikili yüksek olarak Vdd ve ikili düşük olarak referans gerilimlerinin bu düğümde oluşturulması gerekmektedir. Bu da devrede evirici grubu ile sağlanmıştır. Devrenin çalışmasını anlayabilmek için şu iki durum göz önüne alınsın; $0 \leq x < m$ ve $m \leq x < r$.

Durum 1; Eğer $0 \leq x < m$ ise, $I1=xI_0$ ve A ile B düğümlerindeki gerilimler ikili düşüğe seviyede olacaktır. S1 anahtarı kapalı ve S2 anahtarı açık olacaktır. Bu şartlar altında $I2=(r-m)I_0$ ve $I3=0$ olacaktır. Çıkış akımı ise $xI_0+I2-I3$ olacaktır ki bu da $(r-m)I_0$ kadar giriş akımından fazla olacaktır. Yani $x \xrightarrow{m} = x - m - r$ çıkışını elde etmek için giriş akımına $r-m$ mantık seviyesi eklenmiş olmaktadır.

Durum 2; Eğer $m \leq x < r$ ise, $I1=(m-0.5)I_0$ ve A ve B düğümlerindeki gerilimler ikili yüksek seviyede olacaktır. Yani S1 anahtarı açık ve S2 anahtarı kapalı olacaktır. Bu şartlar altında, $I2=0$ ve $I3=mI_0$ olacaktır. Çıkış akımı da giriş akımından mI_0 kadar az olacaktır. Yani $x \xrightarrow{m} = x - m$ olacaktır.

T1, T2 ve T3 transistörlerinin boyutları değiştirilerek değişik m değerleri için devre tasarlanabilir. Aşağıdaki tabloda 4 seviyeli mantık için bu transistörlerin boyutları $m=1$, $m=2$ ve $m=3$ için verilmiştir.

Tablo 4.3 Hibrit Döngü operatörü için transistör boyutları

Döngü operatörü	Transistör boyutları (W/L)		
	T1	T2	T3
$x \xrightarrow{1}$	1/2	3/1	1/1
$x \xrightarrow{2}$	3/2	2/1	2/1
$x \xrightarrow{3}$	5/2	1/1	3/1

Döngü operatörünün eşleniği şöyle tanımlanmıştır;

$$\overline{x^{\leftarrow m}} = (r-1) - x^{\rightarrow r} \quad (4.11)$$

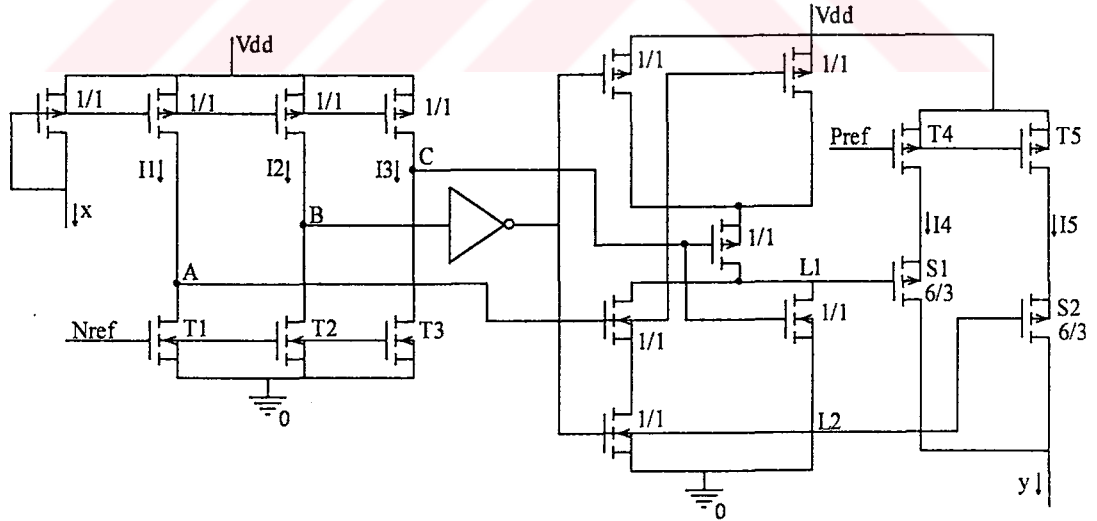
Döngü operatörünün tümleyeni ise aşağıdaki eşitlik kullanılarak gerçekleştirilebilir;

$$x^{\leftarrow m} = x^{\rightarrow -r-m} = x^{\rightarrow -m} \quad (4.12)$$

Döngü veya eşlenik Döngü operatörlerini gerçeklemek için kullanılan transistor sayıları aynıdır.

4.2.1.6. ÇSM akım yenileyici

CMCL devreleri, kendinden yenilemeli (self restoring) değildir. İşaretler değişik yapı blokları arasından dolaşırken bozulmalara uğrayabilmektedir. Dolayısıyla birbirine çok yakın olabilen ÇSM mantık seviyeleri devre tarafından yanlış yorumlanabilmektedir. Bu sorunu aşmak üzere seviye yenileyici kullanmak gerekir. Yenileyici devre örneği aşağıda verilmiştir (4 seviyeli mantık için) [27];



Şekil 4.14 Hibrit ÇSM akım yenileyici

Devre, giriş CMCL katı, evirici, CMOS ikili kapılardan ve CMCL çıkış katından oluşmaktadır. Giriş, evirici ve CMOS kapı kombinasyonu kod çözücü olarak da

kullanılabilir. Bu devre akım modlu ÇSM işaretlerini gerilim modlu ikili işaretlere dönüştürmektedir. Çıkış katı ise tam tersi çalışma mantığı ile kodlayıcı görevi görmektedir.

Giriş katı P tipi akım aynası ve 1,2,3 mantık seviyelerini taramak için 3 adet Thershold elemanından oluşmaktadır. Çıkış katı ise iki anahtar ve iki sabit akım kaynağından oluşmaktadır (constant). Devrede kullanılan transistor boyutları ise; $W1/L1=0.5$, $W2/L2=1.5$, $W3/L3=2.5$, $W4/L4=1$ ve $W5/L5=2$ olarak belirlenmiştir. Devrenin çalışmasını anlamak için şu dört duruma göz atalım; $0 \leq x < 0.5I_0$, $0.5I_0 \leq x < 1.5I_0$, $1.5I_0 \leq x < 2.5I_0$ ve $2.5I_0 \leq x$.

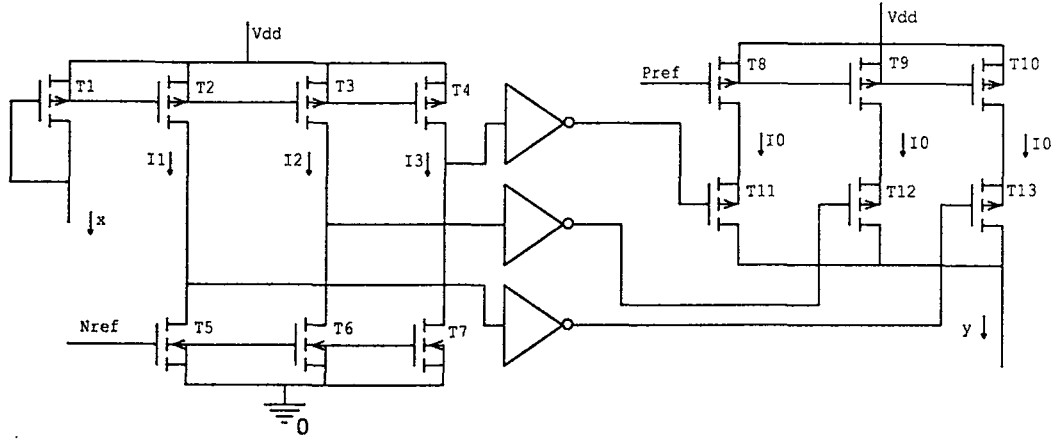
Durum 1; $0 \leq x < 0.5I_0$ ise, $I1=I2=I3=x$ ve A, B ve C düğümlerindeki gerilimler ikili düşük seviyede olacaktır. L1 ve L2 düğümlerindeki gerilim seviyeleri ise ikili yüksek olacaktır. S1 ve S2 anahtarları kapalı olacaktır. $I4=I5=0$ ve çıkış akımı da sıfır olacağından y çıkışı 0 seviyesinde olacaktır.

Durum 2; $0.5I_0 \leq x < 1.5I_0$ ise; $I1=0.5I_0$ ve $I2=I3=x$. A düğümündeki gerilim ikili yüksek seviyede ve B ile C düğümlerindeki gerilimler de ikili düşük seviyede olacaktır. L1 düğümündeki gerilim ikili düşük ve L2 düğümündeki gerilim de ikili yüksek olacaktır. Dolayısıyla S1 açık ve S2 anahtarı kapalı olacaktır. $I4=I_0$ ve $I5=0$ ve çıkış akımı da I_0 olacağından çıkış 1 seviyesinde bulunacaktır.

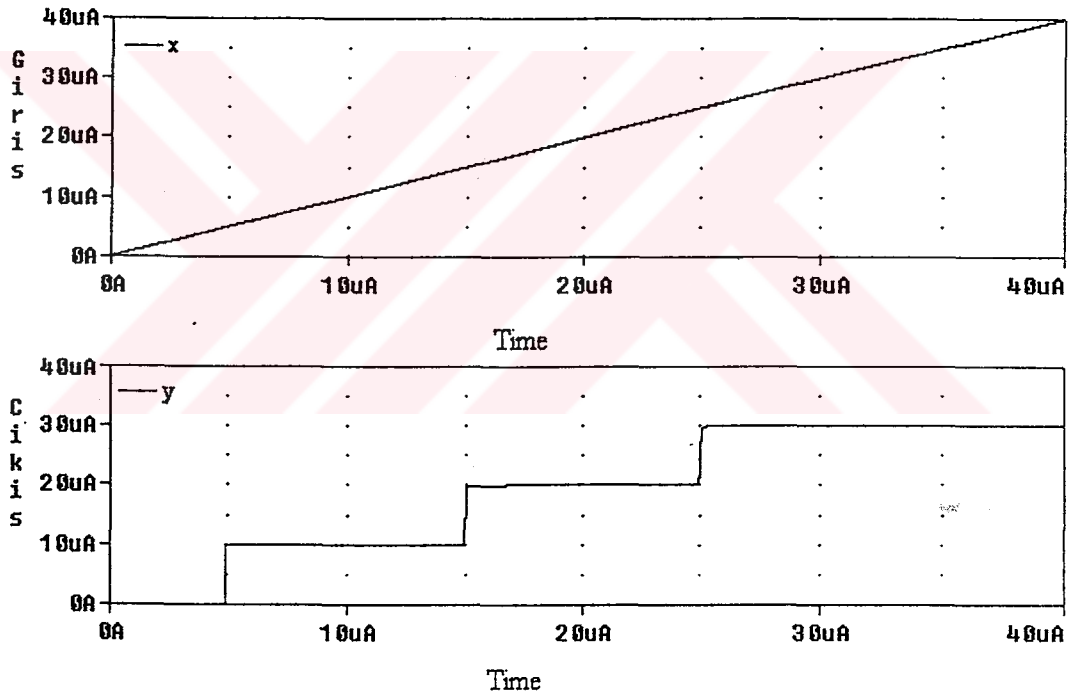
Durum 3; $1.5I_0 \leq x < 2.5I_0$ ise, $I1=0.5I_0$, $I2=1.5I_0$ ve $I3=x$. A ve B düğümlerindeki gerilimler ikili yüksek ve C düğümündeki gerilim düşük olacaktır. L1 düğümündeki gerilim ikili yüksek ve L2 düğümündeki gerilim ikili düşük olacaktır. S1 kapalı ve S2 açık olacaktır. $I4=0$ ve $I5=2I_0$ olacağından çıkış 2 seviyesinde bulunacaktır.

Durum 4; $2.5I_0 \leq x$ ise, $I1=0.5I_0$, $I2=1.5I_0$ ve $I3=2.5I_0$ olacaktır. A, B ve C düğümlerindeki gerilimler ikili yüksek seviyede olacaktır. L1 ve L2 düğümlerindeki gerilimler ise ikili düşük olacaktır. Dolayısıyla S1 ve S2 anahtarları açık olacaktır. $I4=I_0$ ve $I5=2I_0$ olacağından çıkış 3 seviyesinde olacaktır.

Yukarıdaki devrenin geliştirilmiş bir örneği Şekil 4.15'te, simülasyon sonuçları da Şekil 4.16'da verilmiştir [45]. Bu devre, Şekil 4.14'deki devrenin sadece çıkış katı değiştirilerek elde edilmiştir.



Şekil 4.15 Geliştirilmiş Hibrit ÇSM akım yenileyici devresi



Şekil 4.16 Geliştirilmiş Hibrit ÇSM akım yenileyici simülasyon sonuçları

4.2.1.7. İki seviye-dört seviye dönüştürücü (kodlayıcı)

Günümüzde üretimi yapılan ÇSM devrelerinden dışarıya çıkarılan uçlar genellikle gerilim modlu ikili işaretler taşıyacak şekilde tasarlanmaktadır. Bu amaçla kodlayıcı ve kod çözücü devreleri kullanılmaktadır.

Dönüştürücülerin temel ifadesi aşağıdaki gibi tanımlanmıştır [16];

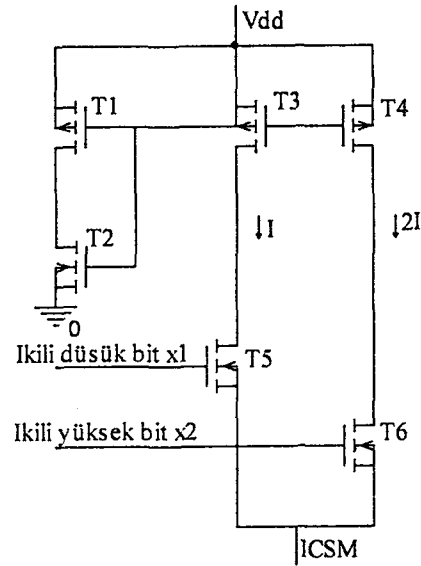
$$S_b = \sum_{i=1}^b 2^{i-1} \cdot x_i \quad (4.13)$$

x_i ikili sayının i 'nci biti, b hane sayısı (digit) ve S_b de ikili ÇSM dönüşümü olarak adlandırılmaktadır. İkili seviye-dört seviye dönüştürücü için x_1 düşük anlamlı hane ve x_2 'de yüksek anlamlı hane olmaktadır.

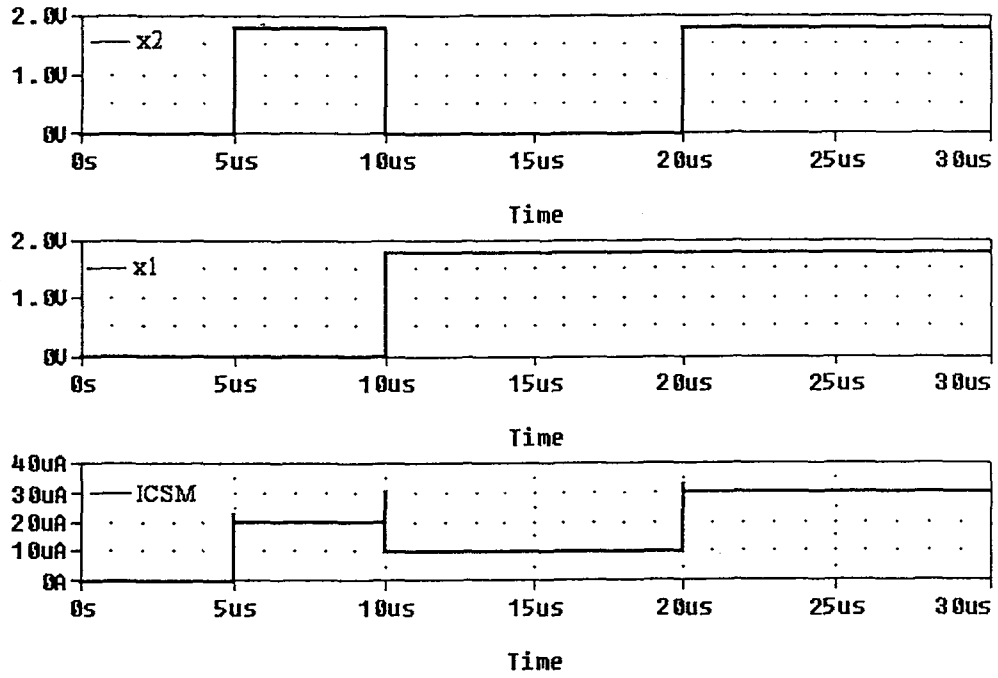
Tablo 4.4 İkili-Dört seviye dönüşümü doğruluk tablosu

İkili		Dört seviyeli
x_2	x_1	
1	1	3
1	0	2
0	1	1
0	0	0

Şekil 4.17'de bir kodlayıcı devresi ve simülasyonu da Şekil.18'de verilmiştir [22];



Şekil 4.17 İkili-Dört seviyeli kodlayıcı



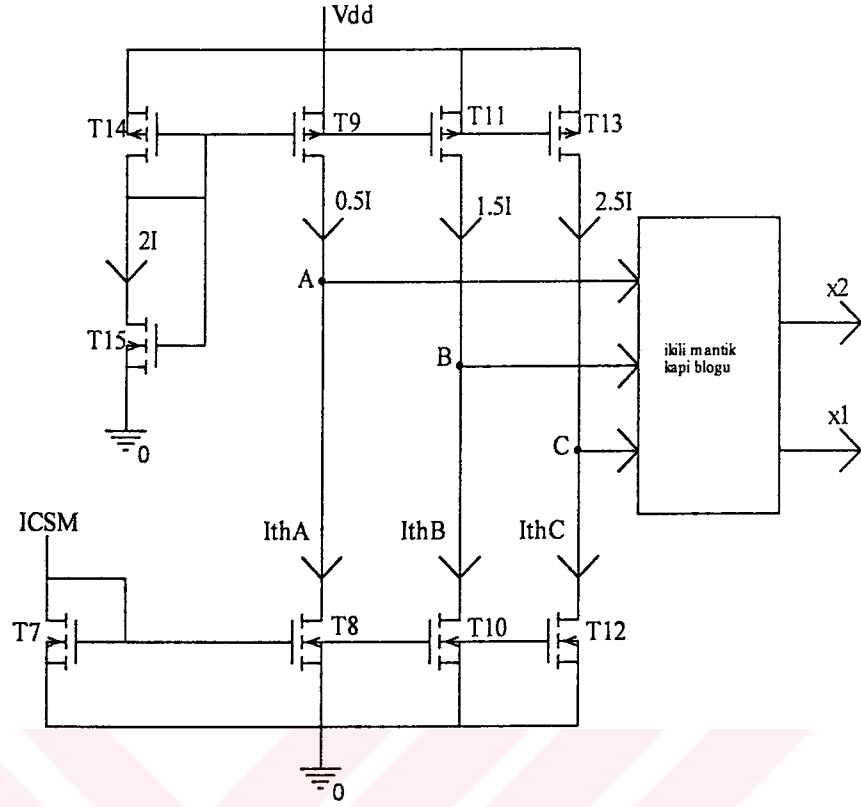
Şekil 4.18 İkili-Dört seviyeli kodlayıcı simülasyon sonuçları

Devrenin çalışma prensibi ise şöyle özetlenebilir; T1-T4 transistor grubu ile referans akımı oluşturulmuş ve kopyalanmıştır. T4 transistorunun akımı T3 transistorunun akımının iki katı olacak şekilde tasarlanmıştır. T5 ve T6 transistorlarına ise ikili gerilim işaretleri uygulanmıştır. Transistorlar basit anahtar olarak çalışmaktadır. Düşük ağırlıklı bit T5 transistoruna ve yüksek ağırlıklı bit de T6 transistoruna uygulanmıştır. T5 ve T6 transistorlarının çıkışı ÇSM akım işaretlerini elde etmek için birleştirilmiştir.

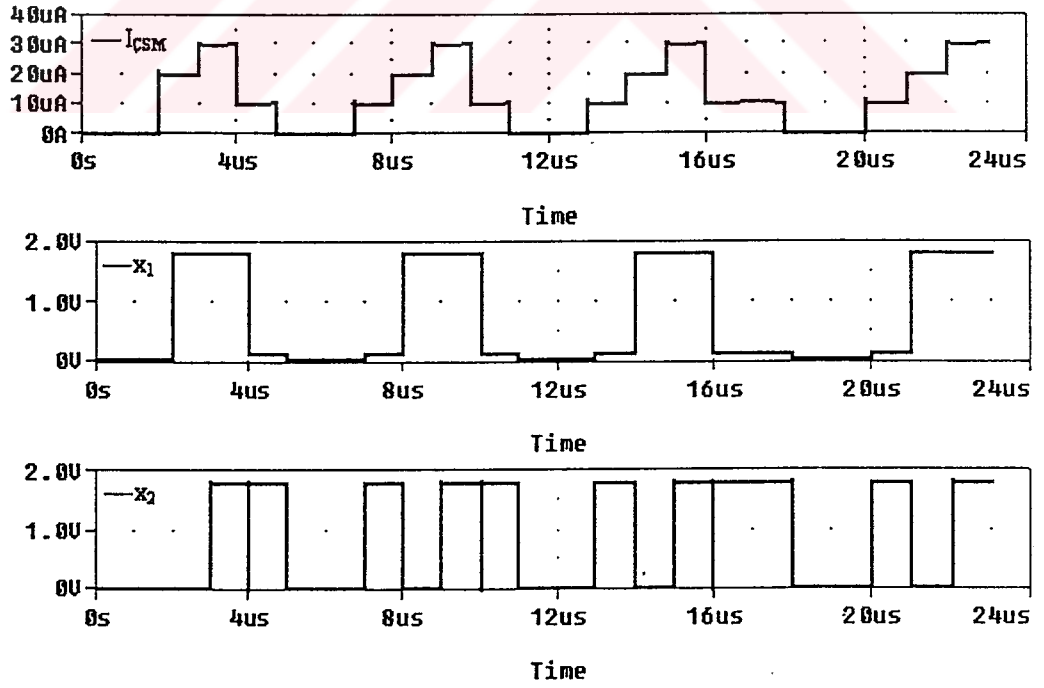
4.2.1.8. Dört-iki seviyeli dönüştürücü (kod çözücü)

Akım modlu 4 seviyeli mantık-gerilim modlu ikili dönüştürücü (kod çözücü) devresi Şekil 4.19'da ve simülasyon sonuçları da Şekil 4.20'de verilmiştir [22]. Devrenin çalışması kısaca aşağıdaki gibi özetlenebilir.

ÇSM akım işaretleri T7 transistorunun savağına uygulanmıştır. T7 transistoru da T8-T9, T10-T11 ve T12-T13 transistor çiftlerinden oluşan akım karşılaştırıcıları sürmektedir. Akım karşılaştırıcıların savak noktaları A, B ve C'deki gerilimler transistorlardan akan akımların referans akımlarından düşük olması halinde ikili yüksek seviyede olacaktır.

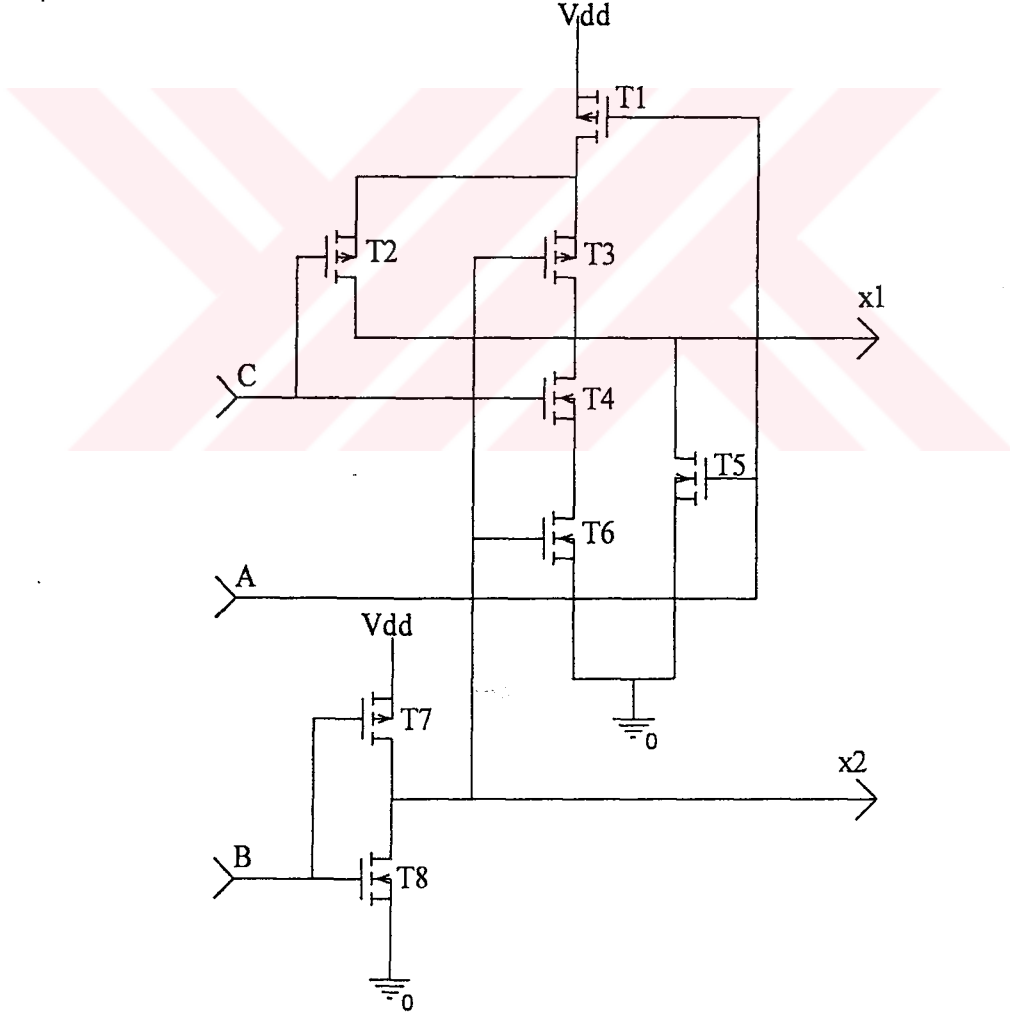


Şekil 4.19 Dört-iki seviyeli kod çözücü devresi



Şekil 4.20 Dört seviyeli ikili kod çözücü simülasyon sonuçları

Giriş akımı $0.5I$ değerini aşınca A düğümlerindeki gerilim 0'a düşecektir, B ve C düğümlerinde gerilim ise yüksek seviyede olacaktır. Benzer olarak giriş akımı $1.5I$ değerini aşınca A ile beraber B düğümündeki gerilimler düşük seviyede olacaktır, C düğümündeki gerilim ise yüksek seviyede kalacaktır. Giriş akımı $2.5I$ akımını aşınca ise her üç düğümdeki gerilim düşük seviyede olacaktır. T14 ve T15 transistörleri ile $2I$ akımı oluşturulmaktadır. Oluşturulan referans akımı W/L oranlarına bağlı olarak T9, T11 ve T13 tarafından 0.25, 0.75 ve 1.25 ile çarpılmaktadır. A, B ve C düğümlerinde oluşturulan gerilimlerle kompleks gerilim modlu ikili devreler sürülmekte ve kod çözme işlemi gerçekleştirilmektedir. Kod çözme işlemini yerine getiren kompleks kapı devrelerinin tasarımı için gereken devre (şekilde ikili mantık) Şekil 4.21'de verilmiştir.



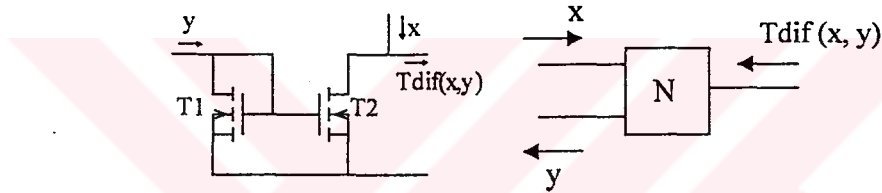
Şekil 4.21 Kod çözücü ikili kapı bloğu devresi

4.2.2. Tam akım modlu devreler

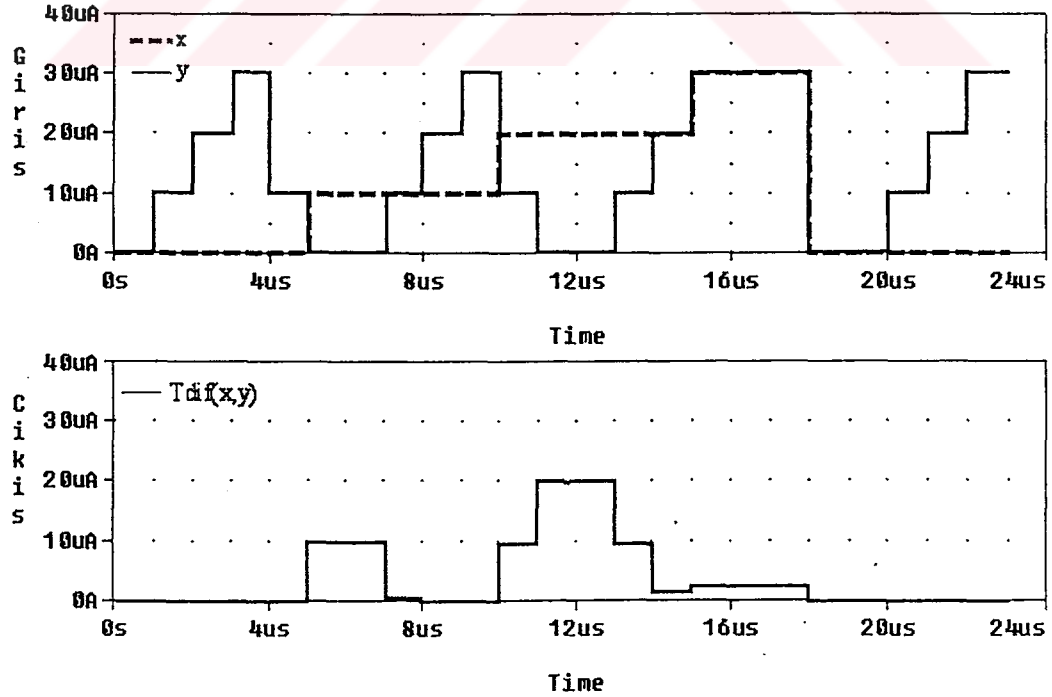
Tamamen akım modlu olarak çalışan ÇSM devre tasarım tekniği, 1987 yılında Onneweer ve Kherkhoff tarafından önerilen sınırlı fark (Tdif) operatörüne dayanmaktadır. Bu operatörün gerçekleştirilmesinde kullanılan örnek bir devre aşağıda verilmiştir. Devre NMOS basit akım aynasından oluşmaktadır. Bu devre tam akım modlu devrelerin temelini oluşturmaktadır [55].

4.2.2.1. Tdif operatörünün devre gerçekleştirilmesi

Tdif operatörünün örnek gerçekleştirilmesi Şekil 4.22’de verilmiştir. Devrede $x > y$ olursa T1 ve T2 transistorlarından y akımı akacaktır. Bu durumda Tdif akımı $x - y$ olacaktır. $x < y$ olursa Tdif akımı 0 olacaktır.



Şekil 4.22 Tam akım modlu Tdif operatörü devresi ve sembolü



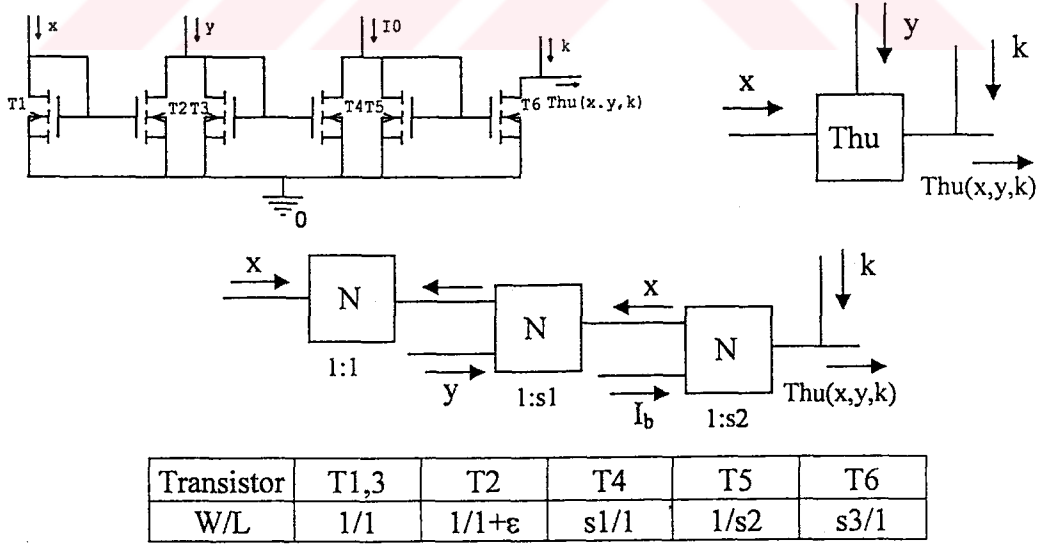
Şekil 4.23 Tam akım modlu Tdif operatörü simülasyon sonuçları

4.2.2.2. Akım çıkışlı akım karşılaştırıcılar

Tam akım modlu devrelerde önemli yapı bloğudur. Kaskad bağlı 3 tane Tdif operatörü devresinden oluşmaktadır. Akım karşılaştırıcılar düşük akım karşılaştırıcı (lower threshold, lower current comparator) ve yüksek akım karşılaştırıcı (upper threshold, upper current comparator) olmak üzere iki tanedir. Yüksek akım karşılaştırıcı devresi Şekil 4.24'te verilmiştir. Düşük akım karşılaştırıcı devresi de aynı olmakla beraber x ve y girişlerini aralarında yer değiştirmek gerekmektedir. Devrede $s1$, $\Delta I = 2,5\mu A$ olmak üzere, $1,5(I_0 / \Delta I)$ olarak seçilmiştir. $s2$ ve $s3$ te $s3 \geq 1,5\sqrt{(r + 0,5)} > s2 > 1$ olarak seçilmiştir. $\varepsilon \geq 0,1$ 'de transistorlardan kaynaklanan hataları gidermek için düzeltme parametresi olarak kullanılmıştır [56]. Tanım ifadeleri aşağıda verilmiştir [31];

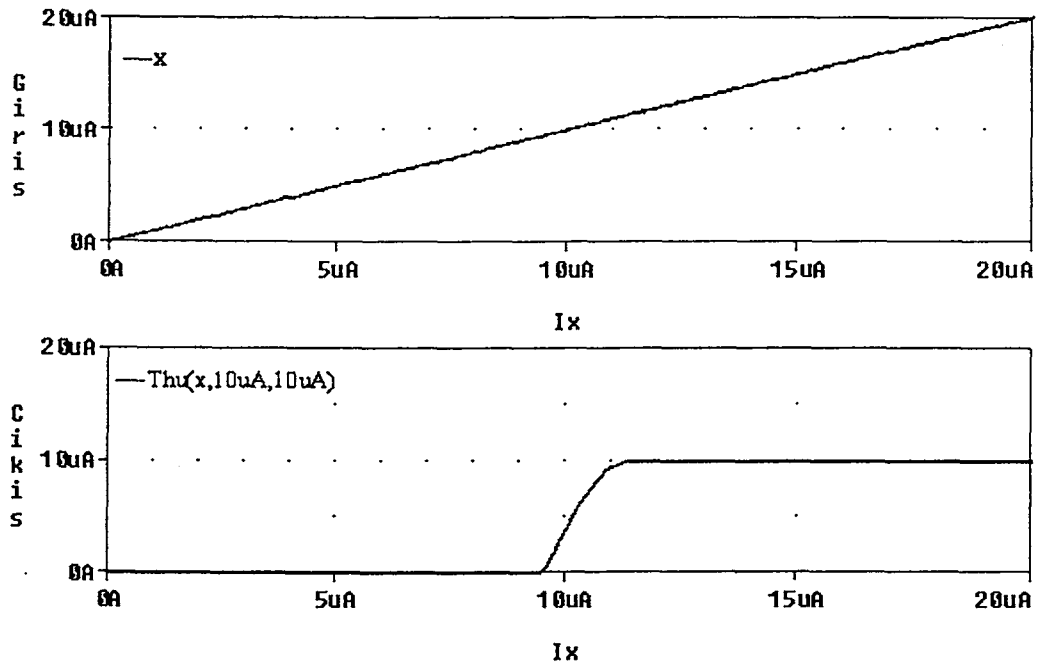
$$Thu(x, y, k) = \begin{cases} k & x \geq y \text{ ise} \\ 0 & \text{aksi halde} \end{cases} \quad (4.14)$$

$$Thl(x, y, k) = \begin{cases} k & x \leq y \text{ ise} \\ 0 & \text{aksi halde} \end{cases} \quad (4.15)$$



Şekil 4.24 Yüksek akım karşılaştırıcı devresi ve sembolü

Yüksek akım karşılaştırıcının simülasyon sonuçları Şekil 4.25'te verilmiştir.



Şekil 4.25 Yüksek akım karşılaştırıcı simülasyon sonuçları

4.2.2.3. ÇSM akım evirici devresinin tasarımı

Akım evirici Tdif operatörü devresinin aynısıdır. Sadece y, x ile yerdeğiştirmekte ve x girişine r-1 mantık değerine karşı düşen akım uygulanmaktadır. Mevcut Literal devreleri ile 0 ve 0'ı içeren Literaller gerçekleştirilememektedir. Bu sorunu aşmak üzere, 0 ve 0'ı içeren Literaller eşlenik Literal olarak gerçekleştirilmektedir. Diğer bir yol da, Literalin çıkışına akım evirici bağlanmaktadır. Evirme işlemi hibrit tasarım mantığında ikili gerilim katında yapılmaktadır. Tam akım modlu devrelerde ise Şekil 4.26'da verilen evirici devresi kullanılmaktadır [56]. Bir x sayısının herhangi bir tabana göre tümleyeni aşağıdaki gibi verilmektedir [28,29];

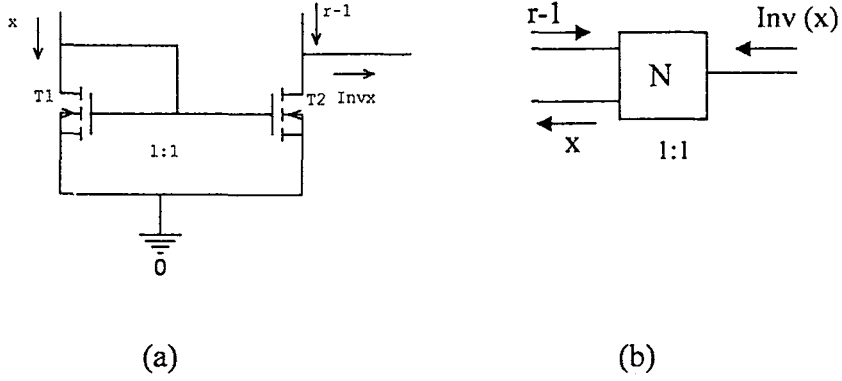
$$\bar{x} = (r - x) - 1 \quad (4.16)$$

Veya

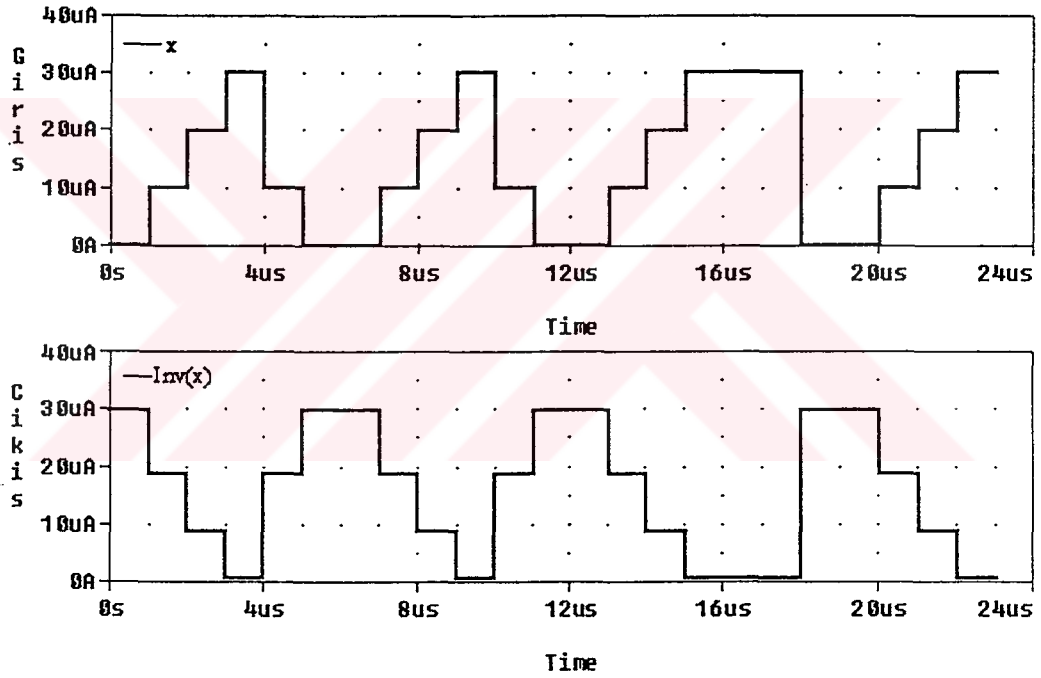
$$\text{Inv}(x) = x^{\overrightarrow{r-1}} \quad (4.17)$$

Şekil 4.26'daki evirici devresinin simülasyon sonuçları Şekil 4.27'de verilmiştir.

Örnek; 4 sayı tabanı için ($r=4$), $x=0 \Rightarrow \bar{x} = 3$ ve $x=2 \Rightarrow \bar{x} = 1$



Şekil 4.26 Akım evirici devresi (a) ve sembolü (b)



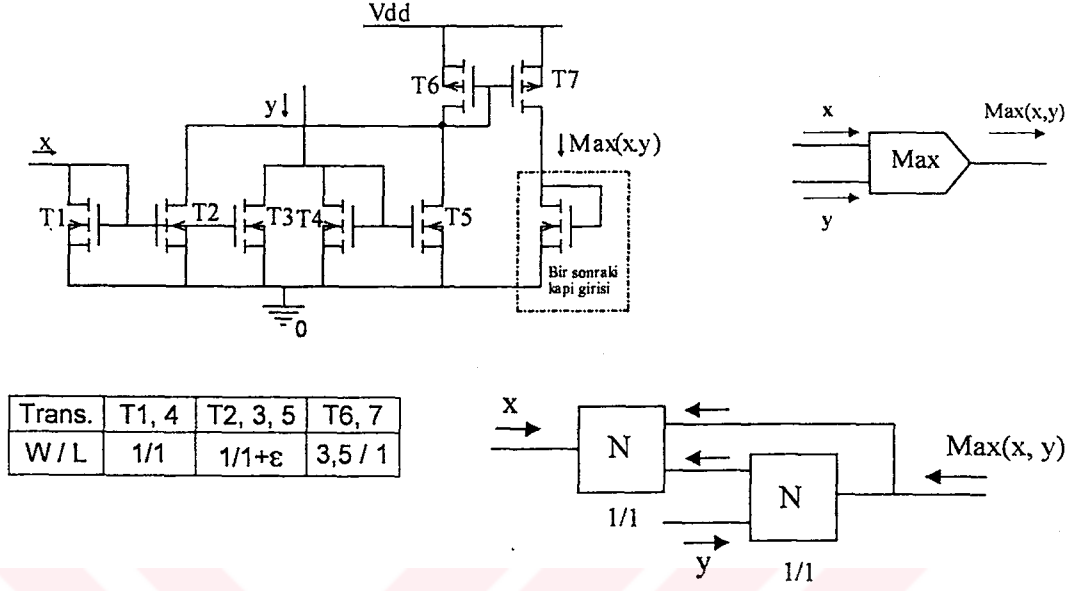
Şekil 4.27 Akım evirici simülasyon sonuçları

4.2.2.4. Max operatörünün devre gerçekleştirilmesi

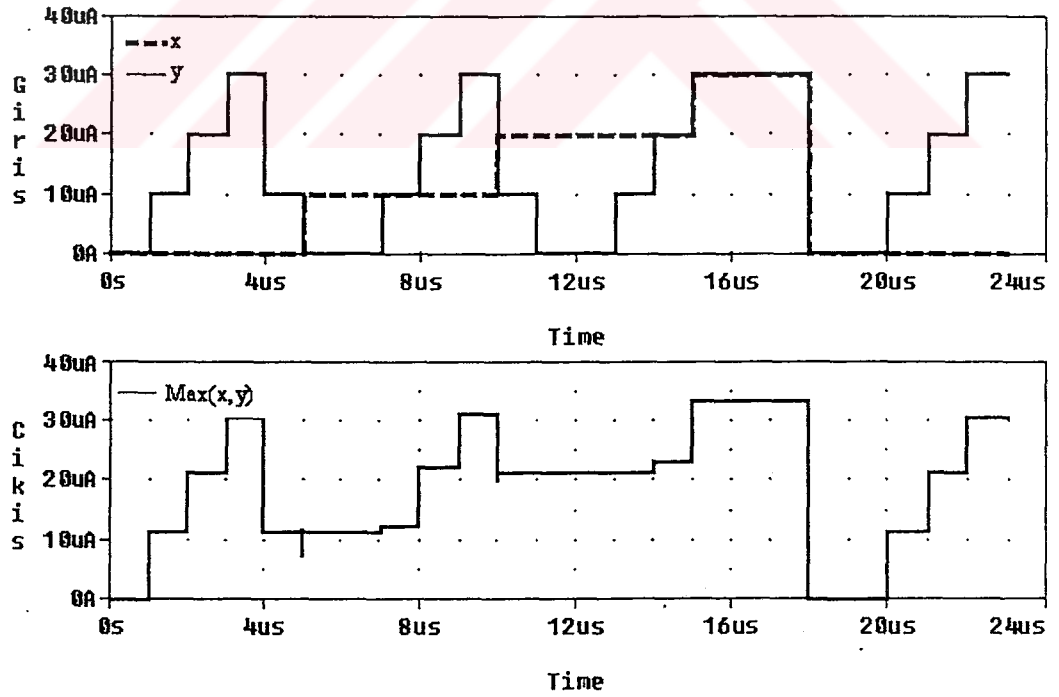
Max operatörünü gerçekleyen devre Tdif operatör topolojisinden yararlanılarak tasarlanmıştır. Max operatörünün Tdif operatörüne bağlı ifadesi aşağıdaki eşitlikteki gibi verilmektedir [56];

$$\text{Max}(x, y) = x - (y \boxminus x) = x \boxplus (y \boxminus x) \quad (4.18)$$

Max operatörünün örnek bir devresi Şekil 4.28’de, simülasyon sonuçları da 4.29’da verilmiştir.



Şekil 4.28 Max operatörü devresi, sembolü ve blok diyagramı

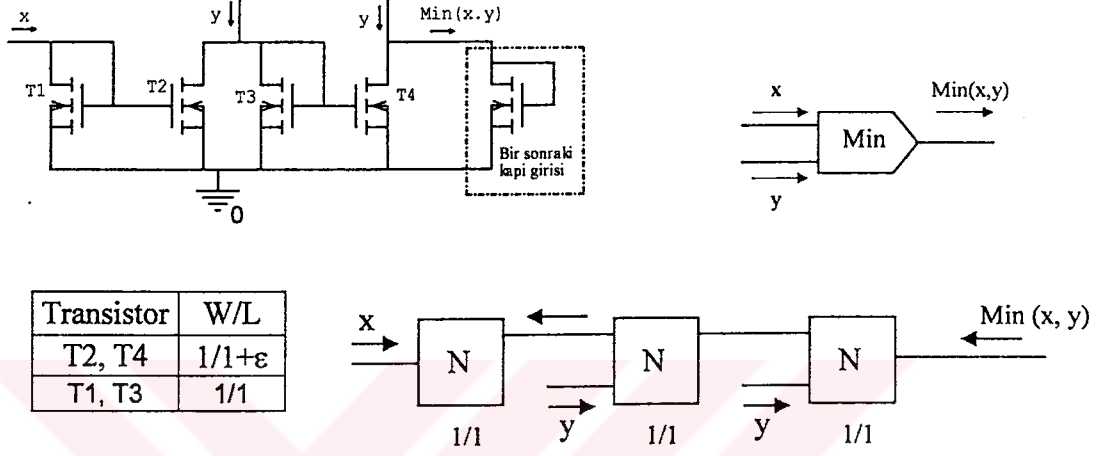


Şekil 4.29 Max operatörü simülasyon sonuçları

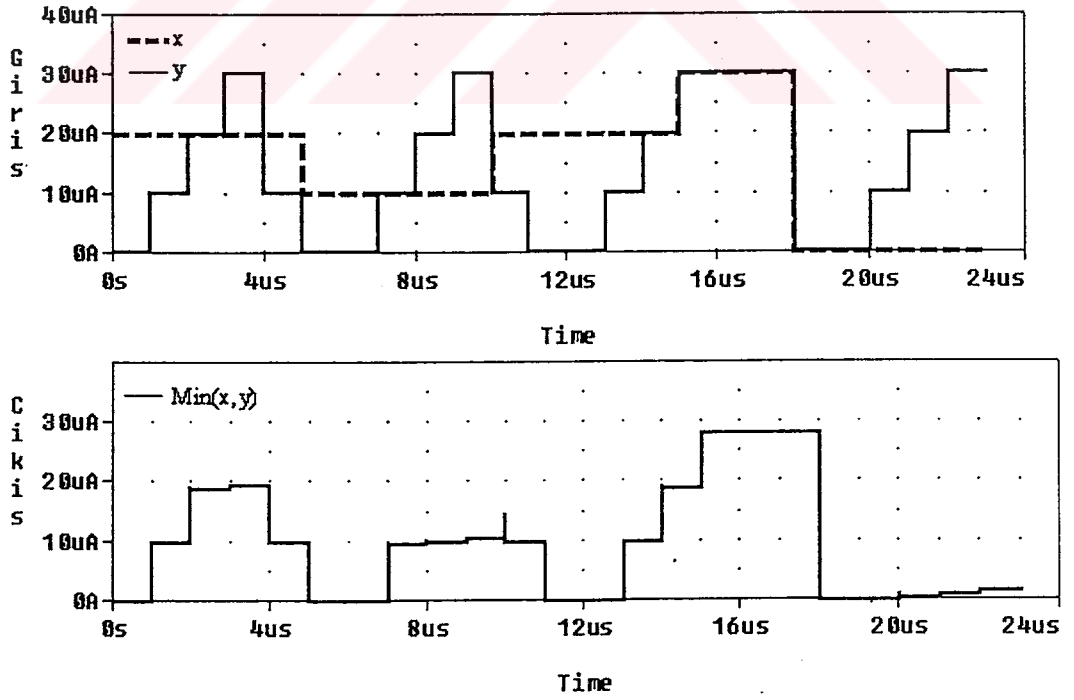
4.2.2.5. Min operatörünün devre gerçekleştirilmesi

Min operatörü ard-arda bağlanmış iki tane Tdif devresinden oluşmaktadır. Bu devre aşağıdaki eşitlikten yararlanılarak oluşturulmuştur [56];

$$\text{Min}(x, y) = x - (x \boxminus y) = x \boxminus (x \boxminus y) \quad (4.19)$$



Şekil 4.30 Min operatörü devresi, sembolü ve blok diyagramı

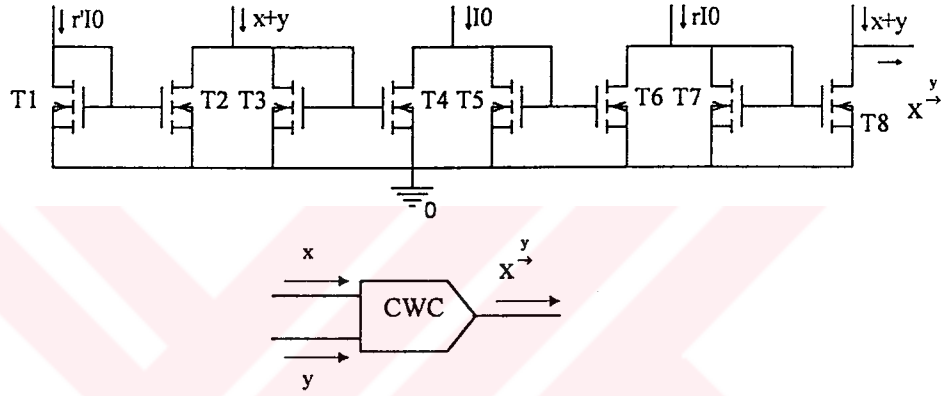


Şekil 4.31 Min operatörü simülasyon sonuçları

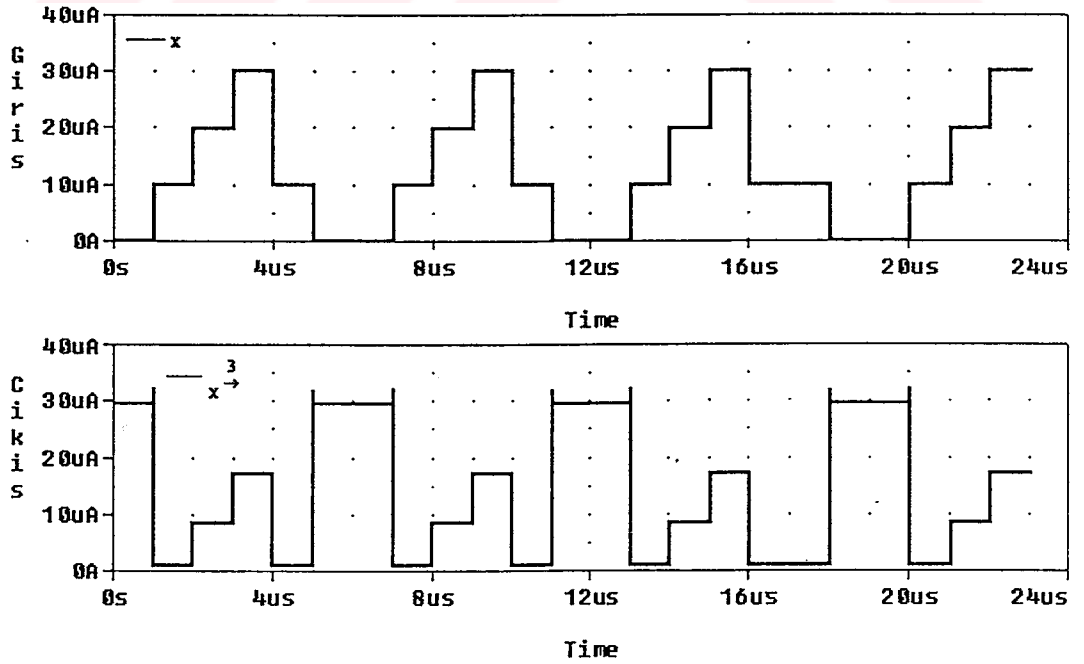
4.2.2.6. Saat yönü Döngü operatörünün devre gerçe klemesi (CWC)

Saat yönü Döngü operatörü'nün Tdif operatörüne ba ğlı genel ifadesi (4.20)'de verilmiştir. Saat yönü Döngü operatörü'nün devre gerçe klemesi de bu ifadeden hareketle Şekil 4.32'deki gibi oluşturulmuştur (transistor boyutları Tdif operatörü ile aynıdır) [57];

$$x \overset{y}{\rightarrow} = (x + y) \boxminus (x + y) \Big|_{r'}^r = (x + y) \boxminus x \Big|_{r-y}^r \quad (4.20)$$



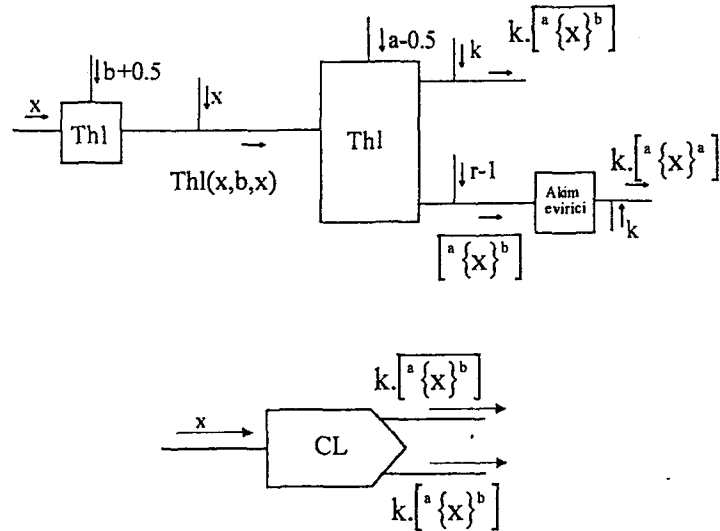
Şekil 4.32 Döngü operatörü devresi ve sembolü



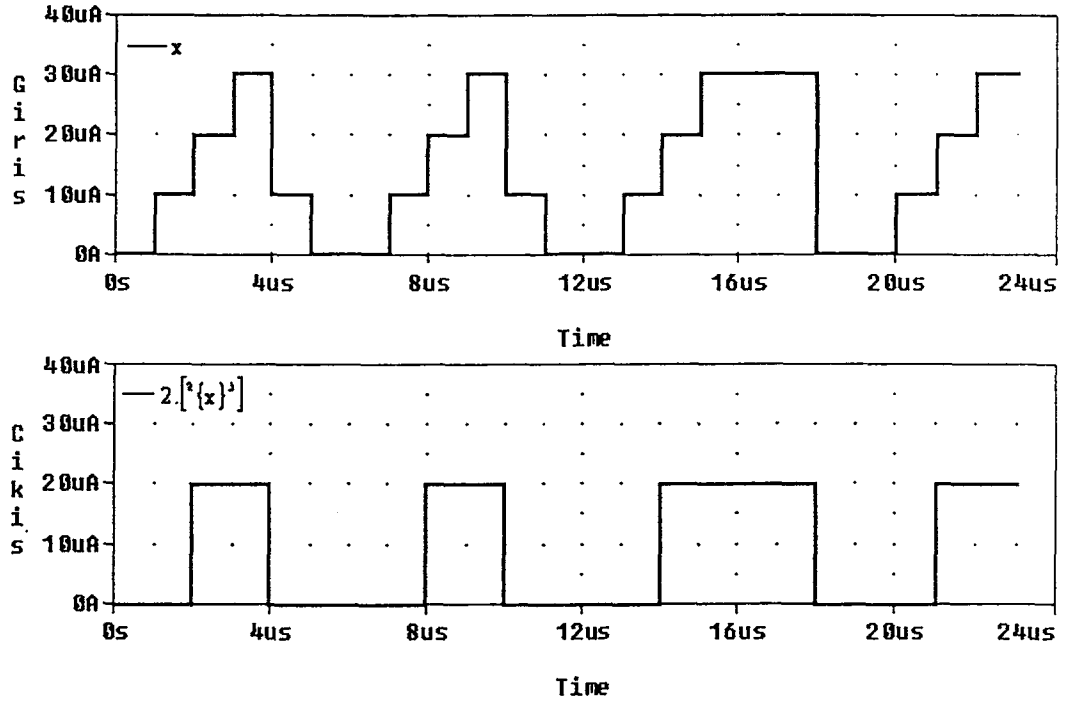
Şekil 4.33 Döngü operatörü simülasyon sonuçları

4.2.2.7. Literal operatörü'nün devre gerçekleştirilmesi

Literal operatörünün devresi ard-arda bağlanmış iki akım karşılaştırıcı ile gerçekleştirilmiştir. Literal operatörü hem yüksek akım hem de düşük akım çıkışlı akım karşılaştırıcı ile gerçekleştirilebilir. Şekil 4.34'te düşük akım karşılaştırıcı ile gerçekleştirilmiş devrenin blok diyagramı verilmiştir. Literal iki mantık seviyesi aralığında çıkış verdiği için önce büyük mantık seviyesi karşılaştırılmakta ve b mantık değerinden küçük akımlar bir sonraki kata iletilmektedir. İkinci katta ise düşük mantık değerinden daha düşük akımların geçişine izin verilmektedir. Yani eşlenik literal gerçekleştirilmektedir. Eşlenik literal işareti de akım evirici kullanılarak Literal operatörü gerçekleştirilmektedir. Yüksek akım karşılaştırıcı ile Literal doğrudan gerçekleştirilebilmektedir. Düşük akım karşılaştırıcı ile aynı devre topolojisi kullanılmaktadır. Sadece a ve b girişlerini kendi aralarında yer değiştirmek gerekmektedir [55]. Literalin üst seviyesi b'nin taranabilmesi için düşük akım karşılaştırıcı girişine b'nin $b+0.5$ olarak uygulanması gerekmektedir. Düşük akım karşılaştırıcının a seviyesinden düşük işaretleri algılayabilmesi için devre girişine a girişinin $a-0.5$ olarak uygulanması gerekmektedir. Zira a girişine $a.I_0$ işareti uygulanırsa karşılaştırıcının bu işareti de algılayıp çıkış verme olasılığı bulunmaktadır. Dolayısıyla literal çıkışı $k.[^{a+1}\{x\}^b]$ olacaktır. Şekil 4.34'te verilen devrenin simülasyon sonuçları Şekil 4.35'te verilmiştir.



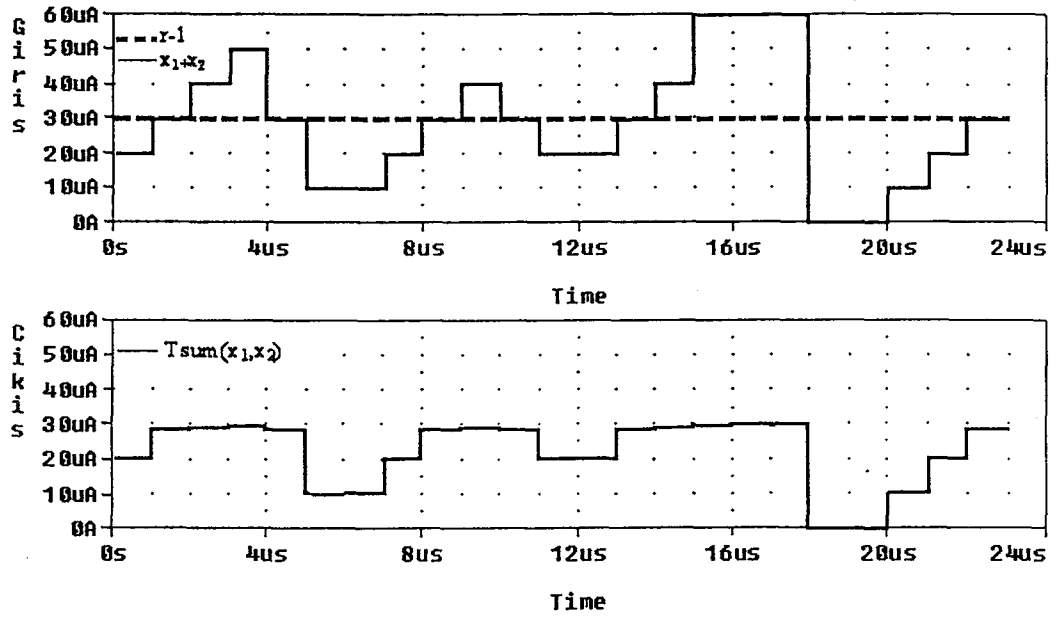
Şekil 4.34 Literal operatörü blok diyagramı ve sembolü



Şekil 4.35 Literal operatörü simülasyon sonuçları

4.2.2.8. Tsum operatörünün devre gerçektelemesi

Tsum operatörü Min operatörüne bağlı olarak (4.21) ifadesinden yararlanılarak gerçekleştirilebilmektedir.



Şekil 4.36 Tam akım modlu Tsum operatörü simülasyon sonuçları

$$Tsum = \text{Min}(x_1 + x_2 + \dots + x_n, r-1) \quad (4.21)$$

Devre yapısı Min operatörünü gerçeklemek için kullanılan devrenin aynısıdır. Tsum operatörü tam akım modlu olarak ilk kez verilmektedir. Devrenin UMC018 parametrelili transistorlar ile benzetimi Şekil 4.36’da verilmiştir. Bu operatör daha önce basit akım aynaları ile tasarlanmıştır. Ancak akım aynaları ile tasarlanan devrede işaret bozulması oldukça yüksektir. Ayrıca klasik tasarımda 9 transistor ile oluşturulan bu operatör devresi 4 transistor ile yeniden tasarlanmıştır.

Hibrit modlu ve tam akım modlu operatör devrelerinin ayrıntılı karşılaştırması için [56,57]’ye bakılabilir.



5. ÖRNEK FONKSİYON TASARIMI

Tablo 5.1’de iki değişkenli ve dört seviyeli fonksiyon verilmiştir. Bu fonksiyonu gerçeklemek için öncelikle seçilecek birleştirici operatöre göre çarpanların toplamı formu elde edilmiştir. Çarpanların toplamı formu elde edildikten sonra ifadeyi indirgeme olasılığı araştırılır. İndirgeme aşamasında, kanonik formda çarpanlar (PT) içerisinde yer alacak operatörler belirlenir. Fonksiyon en basit halini aldıktan sonra, ifadedeki her bir çarpan uygun operatör devresi ile gerçekleştirilir. Eğer aynı girişe ait aynı operatör birden çok PT içerisinde yer alırsa o operatör devrede bir kez yer alır çıkışı akım aynaları ile kopyalanarak diğer PT'lere dağıtılır.

Tablo 5.1 Örnek fonksiyon doğruluk tablosu

$x_2 \backslash x_1$	0	1	2	3
0	1	1	2	3
1	1	0	1	2
2	2	1	1	2
3	0	0	0	1

Bu fonksiyon Max ve Tsum algoritması ile çarpanların toplamı formunda ifade edilsin. Fonksiyonun, Tsum ile çarpanların toplamı ifadesini elde etmek için tablodan Üçüncü bölümde verilen bileşen seçim kriterleri ile öncelikle çarpanlarının belirlenmesi gerekmektedir. Dueck&Miller yöntemi ile çarpanlar seçilsin ve indirgeme yapılsın. Tabloda fonksiyonun Tsum algoritması ile ifade edilmesi durumunda gereken minimum çarpanlar kesikli çizgili şekillerle çevrilmiştir. Buradan hareketle Tsum BOP ile fonksiyonun ifadesi (5.1)’deki gibi olacaktır;

$$f(x_1, x_2) = 1.[^0\{x_1\}^3] \bullet 1.[^0\{x_2\}^0] \boxplus 1.[^2\{x_1\}^3] \bullet 1.[^0\{x_2\}^1] \boxplus 1.[^3\{x_1\}^3] \bullet 1.[^0\{x_2\}^3] \boxplus 1.[^0\{x_1\}^3] \bullet 1.[^2\{x_2\}^2] \boxplus 1.[^0\{x_1\}^0] \bullet 1.[^1\{x_2\}^2] \quad (5.1)$$

Max algoritması ile fonksiyon ifade etmek için sadece asal bileşenlerin yeterli olduğu gösterilmiştir [4]. Örnek fonksiyonun tüm asal bileşenleri aşağıda listelenmiştir;

a) $1.[^0\{x_1\}^3] \bullet 1.[^0\{x_2\}^0]$

b) $1.[^2\{x_1\}^3] \bullet 1.[^0\{x_2\}^2]$

c) $1.[^3\{x_1\}^3] \bullet 1.[^0\{x_2\}^3]$

d) $1.[^0\{x_1\}^3] \bullet 1.[^2\{x_2\}^2]$

e) $1.[^0\{x_1\}^0] \bullet 1.[^0\{x_2\}^2]$

f) $2.[^2\{x_1\}^3] \bullet 2.[^0\{x_2\}^0]$

g) $2.[^0\{x_1\}^0] \bullet 2.[^2\{x_2\}^2]$

h) $2.[^3\{x_1\}^3] \bullet 2.[^0\{x_2\}^2]$

i) $3.[^3\{x_1\}^3] \bullet 3.[^0\{x_2\}^0]$

Max algoritması ile fonksiyonun ifadesi (5.2)'de verilmiştir. Şekil 4.28'de verilen Max operatörünün sadece iki girişi bulunmaktadır. Dolayısıyla (5.2) ifadesindeki fonksiyonu gerçeklemek için sekiz adet Max operatörü gerekmektedir. Bu kadar çok Max operatörü kullanmamak için (5.2) ifadesi düzenlenerek (5.3)'te verilmiştir.

$$f(x_1, x_2) = 1.[^0\{x_1\}^3] \bullet 1.[^0\{x_2\}^0] \oplus 1.[^2\{x_1\}^3] \bullet 1.[^0\{x_2\}^2] \oplus 1.[^3\{x_1\}^3] \bullet 1.[^0\{x_2\}^3] \oplus$$

$$\oplus 1.[^0\{x_1\}^3] \bullet 1.[^2\{x_2\}^2] \oplus 1.[^0\{x_1\}^0] \bullet 1.[^0\{x_2\}^2] \oplus 2.[^2\{x_1\}^3] \bullet 2.[^0\{x_2\}^0] \oplus \quad (5.2)$$

$$\oplus 2.[^0\{x_1\}^0] \bullet 2.[^2\{x_2\}^2] \oplus 2.[^3\{x_1\}^3] \bullet 2.[^0\{x_2\}^2] \oplus 3.[^3\{x_1\}^3] \bullet 3.[^0\{x_2\}^0]$$

$$f(x_1, x_2) = (1.[^0\{x_1\}^3] \bullet 1.[^0\{x_2\}^0] + 1.[^2\{x_1\}^3] \bullet 1.[^0\{x_2\}^2] + 1.[^3\{x_1\}^3] \bullet 1.[^0\{x_2\}^3] +$$

$$+ 1.[^0\{x_1\}^3] \bullet 1.[^2\{x_2\}^2] + 1.[^0\{x_1\}^0] \bullet 1.[^0\{x_2\}^2]) \oplus (2.[^2\{x_1\}^3] \bullet 2.[^0\{x_2\}^0] + \quad (5.3)$$

$$+ 2.[^0\{x_1\}^0] \bullet 2.[^2\{x_2\}^2] + 2.[^3\{x_1\}^3] \bullet 2.[^0\{x_2\}^2]) \oplus 3.[^3\{x_1\}^3] \bullet 3.[^0\{x_2\}^0]$$

İfade + lineer toplamı göstermektedir. (5.3) ifadesini gerçeklemek için sadece iki Max operatörü gerekmektedir. Ancak ifadeye dikkat edilirse, a) ve e) PT'lerinin her ikisi de $(x_1, x_2) = (0, 0)$ noktasına 1 seviye ile katkıda bulunmaktadır. Dolayısıyla bu işaretlerin lineer toplanması ile $(0, 0)$ noktasında 2 seviyeli işaret elde edilmiş olacaktır. Aynı noktayı iki defa gerçeklememek için PT'lerdeki literallerin sınırlarını daraltmak gerekmektedir. Literallerin sınırlarının daraltılması ile elde edilen yeni ifade (5.4)'te verilmiştir.

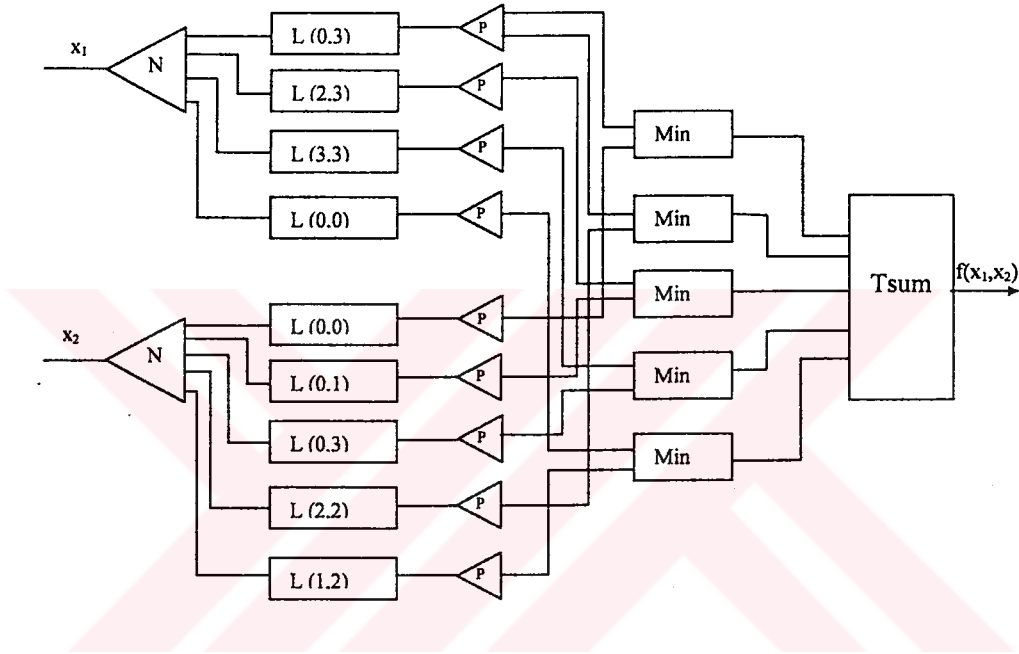
$$f(x_1, x_2) = (1.[^0\{x_1\}^1] \bullet 1.[^0\{x_2\}^0] + 1.[^2\{x_1\}^2] \bullet 1.[^1\{x_2\}^1] + 1.[^3\{x_1\}^3] \bullet 1.[^3\{x_2\}^3] +$$

$$1.[^1\{x_1\}^2] \bullet 1.[^2\{x_2\}^2] + 1.[^0\{x_1\}^0] \bullet 1.[^1\{x_2\}^1]) \oplus (2.[^2\{x_1\}^3] \bullet 2.[^0\{x_2\}^0] + \quad (5.4)$$

$$2.[^0\{x_1\}^0] \bullet 2.[^2\{x_2\}^2] + 2.[^3\{x_1\}^3] \bullet 2.[^1\{x_2\}^2]) \oplus 3.[^3\{x_1\}^3] \bullet 3.[^0\{x_2\}^0]$$

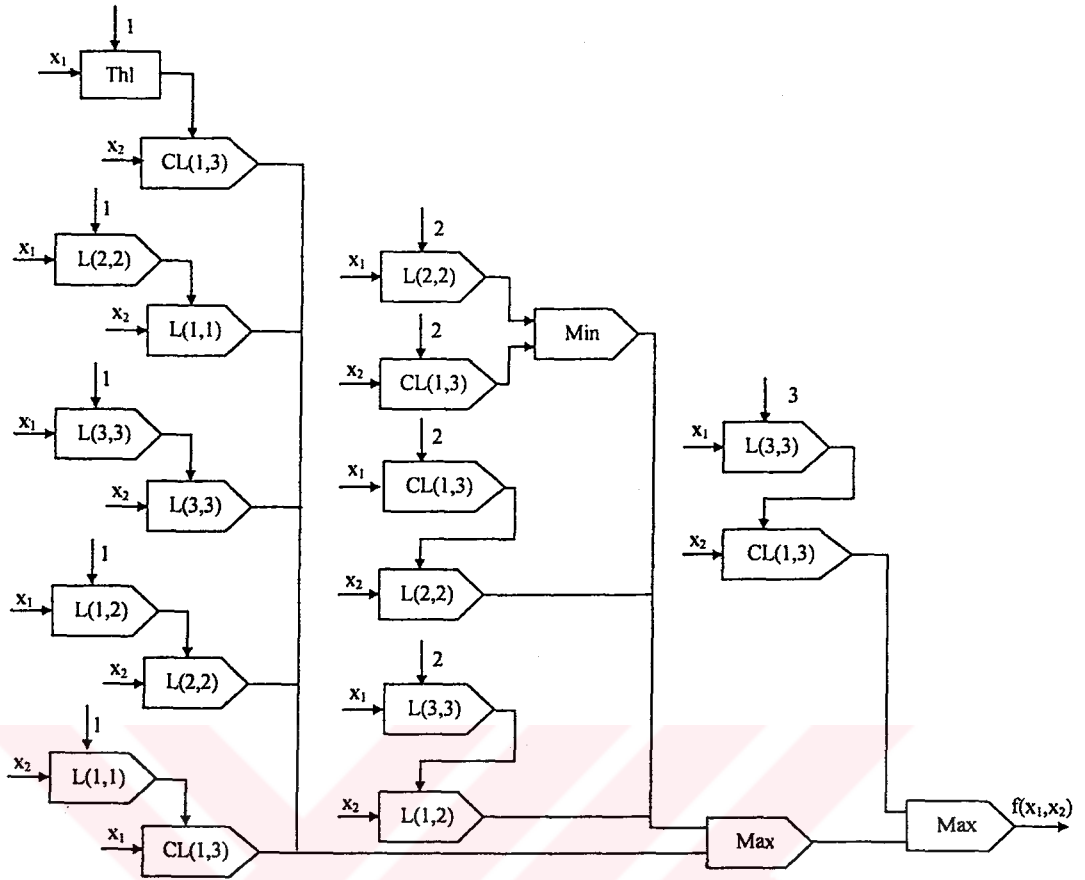
Çarpanları gerçeklemek için Literal operatörü seçilmiştir. Fonksiyonun Tsum algoritmasına göre gerçekleştirilmesi hibrit devre blokları ile Şekil 5.1'de verilmiştir. Benzer devrelerin simülasyon sonuçları için [27]'ye bakılabilir. $1.[^0\{x_1\}^3]$ Literali iki

PT içerisinde yer aldığından, Literal tek bir defa gerçekleşmiş ve uygun akım aynası ile çıkış işareti kopyalanmıştır. $1.\left[\begin{smallmatrix} 0 \\ \{x_i\}^3 \end{smallmatrix}\right]$ Literalinin dikkat edilirse Min fonksiyonunun özelliği dolayısıyla devrede yer almasına gerek yoktur. Tablo 4.2'den görüleceği üzere bu Literal sabit akım kaynağı ile gerçekleştirilebilmektedir ve girişten bağımsızdır. 0 içeren diğer Literaller de tablodan hareketle eşlenik olarak gerçekleştirilmiştir. Şekil 5.1,2,4'te $L(a,b) \left[\begin{smallmatrix} a \\ \{x\}^b \end{smallmatrix}\right]$ 'yi temsil etmektedir. Benzer olarak eşlenik literal de $CL(a,b)$ olarak verilmiştir.

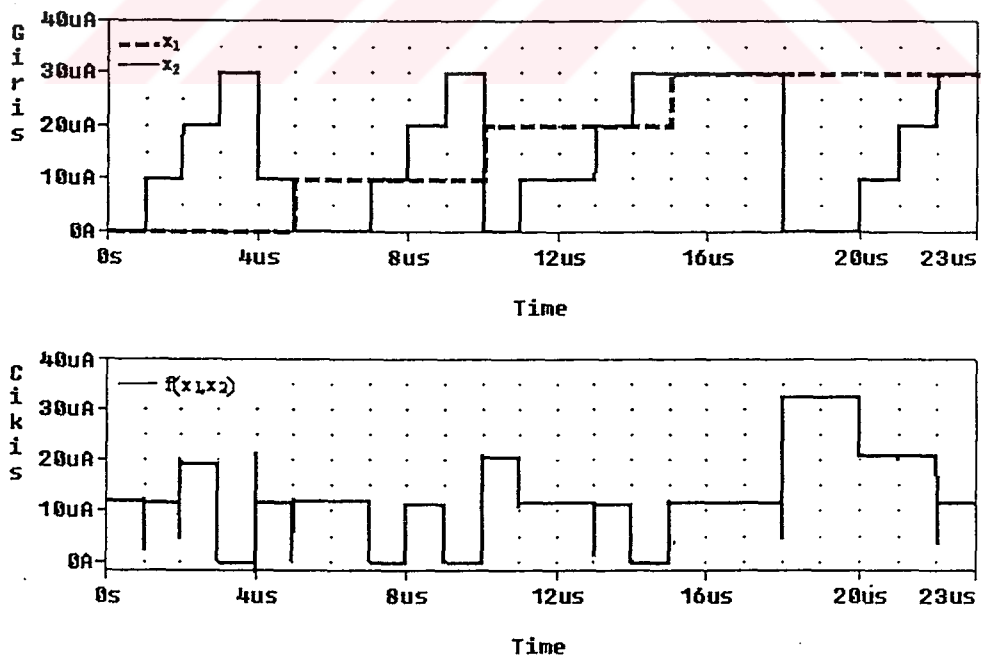


Şekil 5.1 Hibrit Tsum algoritması ile örnek fonksiyonun blok diyagramı

Örnek fonksiyonun tam akım modlu devrelerle Max (ifade (5.4)) algoritması ile gerçekleştirilmiş örnek bir devresi şematik olarak Şekil 5.2'de verilmiştir. Akım modlu devrelerde akım çıkışlı akım karşılaştırıcı veya Literal kaskad bağlanarak Min operatörü elimine edilebilmektedir. Min operatörü ile de kurulan bir PT şekil içerisinde yer almaktadır. Hibrit devrelerde olduğu gibi 0 ve 0'ı içeren Literaller Literal devresi ile gerçekleştirilememektedir. $\left[\begin{smallmatrix} 0 \\ \{x\}^i \end{smallmatrix}\right]$ ($i \neq 0$) Literalleri eşlenik Literal ile $\left[\begin{smallmatrix} i+1 \\ \{x\}^{r-1} \end{smallmatrix}\right]$ olarak veya düşük akım karşılaştırıcı ile $Thl(i,1,1)$ olarak gerçekleştirilmektedir. 0 Literalı tam akım modlu devre tasarım yaklaşımı ile $\left[\begin{smallmatrix} 1 \\ \{x\}^{r-1} \end{smallmatrix}\right]$ olarak gerçekleştirilebilmektedir.

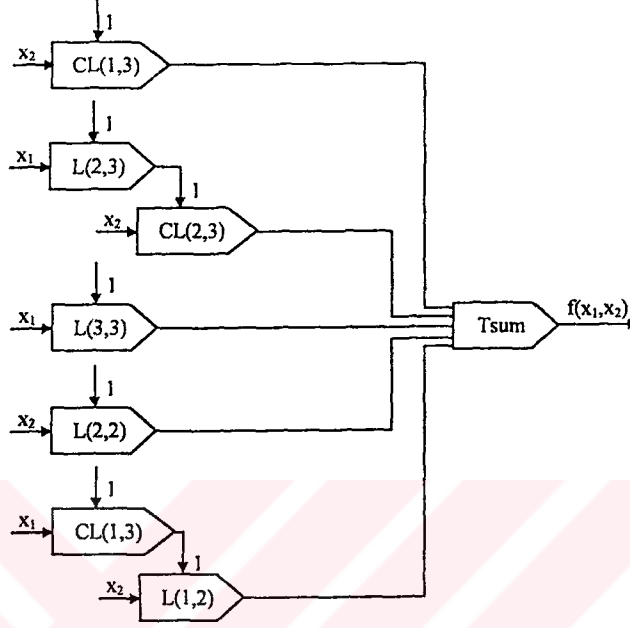


Şekil 5.2 Max algoritması ile örnek fonksiyonun blok diyagramı



Şekil 5.3 Örnek fonksiyonun Max gerçekleştirilmesinin simülasyonu

Tam akım modlu devrelerle fonksiyonun Tsum operatörü ile gerçekleştirilmesi şematik olarak Şekil 5.4'te verilmiştir. Devreden görüldüğü üzere kullanılan operatör bloğu sayısı Şekil 5.2'deki devreye göre yarı yarıya azalmaktadır. Devredeki transistor sayısının azaltılması işaret bozulmasını azaltacaktır.



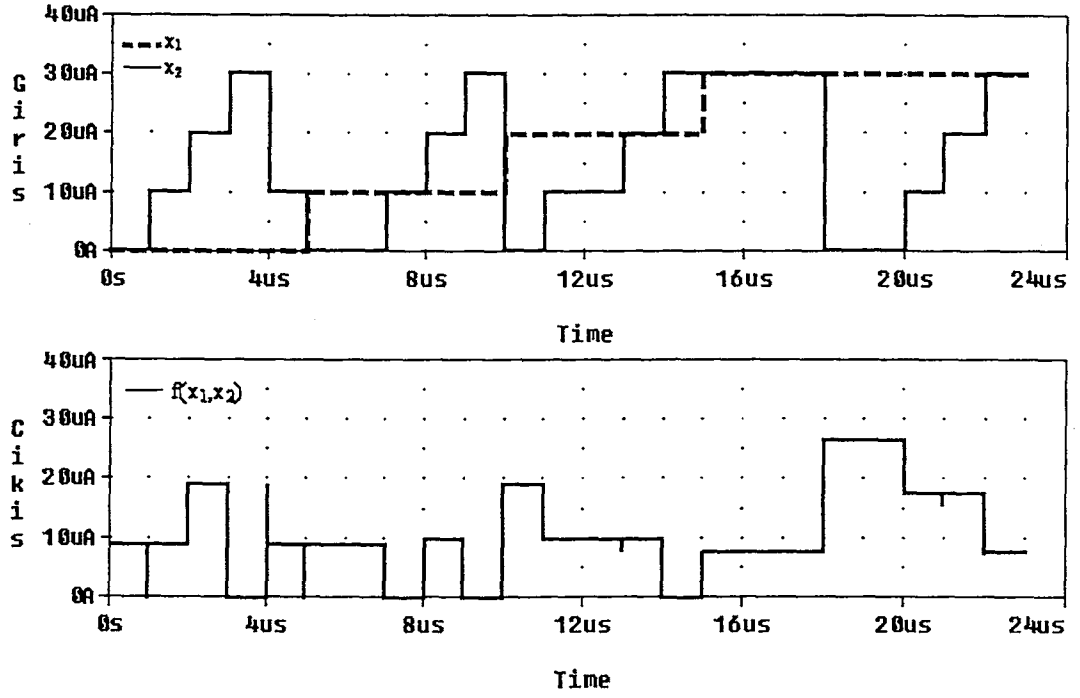
Şekil 5.4 Tsum algoritması ile örnek fonksiyonun blok diyagramı

Şekil 5.2'de blok diyagramı verilen devrenin simülasyon sonuçları Şekil 5.5'te verilmiştir. Şekilden görüldüğü üzere işaret bozulması kabul edilebilir sınırlar içerisinde. Zira tam akım modlu devre tasarım tekniğinde işaret seviyesini algılamak için kullanılan akım çıkışlı akım karşılaştırıcılarda işaret tarama aralığı aşağıdaki ifadedeki gibi tanımlanmaktadır;

$$I_m = [-0.5I_0 + I_i + 0.5I_0] \quad (5.5)$$

İfade (5.5)'de I_0 taban akımı, I_i herhangi bir mantık seviyesine karşı gelen akımı ve I_m de I_i akım seviyesine karşı gelen mantık seviyesini belirtmektedir. $m, i \in r$. Örneğin taban akımı $10\mu A$ olan bir sistemde $5\mu A - 15\mu A$ aralığındaki işaretler mantık 1 olarak algılanacaktır.

Örnek fonksiyonun, üç değişik algoritmaya göre Şekil 5.1,2,4'te verilen tasarımlarının kısa karşılaştırması Tablo 5.2'de verilmiştir.



Şekil 5.5 Örnek fonksiyonun Tsum gerçekleştirilmesinin simülasyonu

Tablo 5.2 Tasarım algoritmalarının karşılaştırılması

Tasarım yöntemi	Kullanılan operatör-temel devre sayısı	Kullanılan transistör sayısı	Ortalama Gecikme (ns)
Hibrit Tsum	19	137	---
Tam akım modlu Max	20	248	10.3
Tam akım modlu Tsum	8	96	8

6. SONUÇLAR VE DEĞERLENDİRME

Çok seviyeli mantığın, tümüyle ikili mantığın yerini alması beklenmemektedir. Fiziksel veya mali kısıtlamalar sebebiyle transistör boyutlarındaki küçülme hızı yavaşlamıştır. Oysa hızlı ve karmaşık işlemleri yapan devrelere olan ihtiyaç sürekli artmaktadır. Buna karşıt ikili mantık devrelerinin tasarımı giderek zorlaşmaktadır. Çok seviyeli mantığın getirdiği “fonksiyonel esneklik” sayesinde birçok mantık işleminin daha ucuza mal edilebileceği bilinmektedir.

Bu çalışmada, tam akım modlu Min operatöründen yeni Tsum operatörü tasarlanmıştır. Bu operatörün indirgemeye daha elverişli olduğu, gerçekleşmesi için klasik örneklerine göre daha az transistör gerektirmesi, tüm fonksiyonun gerçekleşmesinde kullanılan transistör sayısının en az %30 azaltılabileceği gösterilmiştir. Daha önce tasarlanan kodlayıcı/kod çözücü, seviye yenileyici devrelerinin simülasyonu ilk kez verilmiştir.

Bu belge, CMOS ÇSM akım modlu devre tasarımı konusunda çalışmayı düşünecek kişilere referans olması amacıyla, mevcut devre tasarım tekniklerini ve devrelerin çalışma prensiplerini ayrıntılı olarak içermektedir. Bu aşamadan sonra, mevcut devrelerin basitleştirilmesi, teorik olarak mevcut olan ancak CMOS teknolojisi ile tasarlanmamış yeni devrelerin tasarımına odaklanılmasına olanak sağlanmıştır.

ÇSM'nin düşük maliyet, fonksiyonel esneklik gibi güçlü yanları bulunmasına rağmen hala oldukça ciddi sorunları bulunmaktadır. Özellikle akım modlu devrelerde statik güç tüketimi hala yüksek mertebededir. İndirgeme yöntemleri de kabul edilebilir çözümler üretememektedir. Bilgisayara uyarlanabilen basit indirgeme yöntemlerinin geliştirilmesi ÇSM'ye önemli katkı sağlayacaktır. Ayrıca ÇSM'de imkan dahilinde olmasına rağmen tanımlanan ve tasarlanan operatör/devre sayısı oldukça kısıtlıdır. Az elemanlı ve matematik modellemeye uygun olarak tasarlanacak devreler ÇSM'nin fonksiyonel esnekliğini arttıracaktır.

KAYNAKLAR

- [1] **Etiemble D. and Israel M.**, 1988. Comparison of binary multivalued ICs according to VLSI criteria, *IEEE Computer*, 21, 28-42.
- [2] **Smith K.C.**, 1981. The prospects for multiple valued logic; A Technology and Applications View, *IEEE Trans. Computer*, C-30, 619-632.
- [3] **Current K.W.**, 1989. Application of quaternary logic to design of a proposed discrete cosine transform chip, *International Journal of Electronics*, 67 no.5, 687-701.
- [4] **Tirumalai P.P. and Buttler J.T.**, 1989. Prime and non-prime implicants in the minimization of multiple valued logic, *Proceedings of 19th International Symposium on Multiple-Valued Logic (ISMVL)*, China, 272-279.
- [5] **Hurst S.L.**, 1984. Multiple-Valued Logic and Its Status and Its Future, *IEEE Trans. Computers*, C-33, 1160-1179.
- [6] **Lee C.Y. and Chen W.H.**, 1956. Several-valued combinational switching circuit, *Trans. AIEE (Communication electronics)*, 75, 278-283.
- [7] **Allen C.M. and Givone D.D.**, 1968. A minimization technique for multiple-valued logic systems, *IEEE Trans. Computers*, C-17, 182-184.
- [8] **Smith K.C.**, 1988. Multiple-valued logic – A tutorial and appreciation, *IEEE Computer*, 21, 17-27.
- [9] **Miller D.M.**, 1979. On the minimization of many-valued functions, *Proc. ISMVL*, 294-299.
- [10] **Pomper G. And Armstrong J.A.**, 1981. Representation of multivalued functions using the direct cover algorithm, *IEEE Trans. Computers*, C-30, 674-679.
- [11] **Besslich P.W.**, 1986. Heuristic minimization of MVL functions; A direct cover approach, *IEEE Trans. On Computers*, C-35, 134-144.
- [12] **Dueck G.W. and Miller D.M.**, 1987. A direct cover MVL minimization using the truncated sum, *Proc. 17th ISMVL*, 221-227.

- [13] **Druzeta A., Vranesic Z.G. and Sedra A.S.**, 1974. Application of multithreshold elements in the realization of many-valued logic networks, *IEEE Trans. On Computers*, **C-23**, 1194-1198.
- [14] **Alioto M. and Palumbo G.**, 1999. CML and ECL; Optimized Design and Comparison, *IEEE Trans. on Circuits and Systems*, **46**, 1330-1341.
- [15] **Boyle W.S and Smith G.E.**, 1970. Charge-coupled semiconductor devices, *Bell Syst. Tech. J.*, **49**, 587-593.
- [16] **Kerkhoff H.G. and Tervoert M.L**, 1981. Multiple-Valued Logic Charge-Coupled Devices, *IEEE Trans. on Computers*, **C-30**, 644-652.
- [17] **Dao T.T., McCluskey E.J. and Russel R.K.**, 1977. Multivalued integrated injection logic, *IEEE Trans. on Computers*, **C-26 no.12**, 1233-1241.
- [18] **Silio C.B., Pugsley J.H. and Jeng B.A.**, 1979. Control memory reduction using multivalued ROM's, *Proc. IEEE ISMVL*, 19-26.
- [19] **Yamada M., Fujishima K., Nagasawa K. and Gamou Y.**, 1978. A new multilevel storage structure for high density CCD memory, *IEEE J. Solid-State Circuits*, **CS-13**, 688-692.
- [20] **Makino H., Nakase Y., Suzuki H., Morinaka H., Shinohara H. and Mashino K.**, 1996. An 8,8-ns 54*54-Bit Multiplier with High Speed Redundant Binary Architecture, *IEEE J. of Solid State Circuits*, **31 no 6**, 773-782.
- [21] **Kameyama M., Kawahito S. and Higuchi T.**, 1988. A multiplier chip and multiple-valued bidirectional current-mode logic, *IEEE Computer*, **21**, 43-56.
- [22] **Current K.W.**, 1994. Current-Mode CMOS Multiple-Valued Logic Circuits, *IEEE J. of Solid-State Circuits*, **29 no 2**, 95-107.
- [23] **Dao T.T, Davio M. and Gossart G.**, 1980. Complex number arithmetic with odd valued logic, *IEEE Trans. on Computers*, **C-26**, 604-610.
- [24] **Khekhoff H.G., Tervoert M.L. and Stemerdink J.A**, 1981. The design and application of a CCD four-valued full adder circuit, *IEEE Dig. COMPCON*, 96-99.
- [25] **Freitas D.A. and Current K.W.**, 1983. A quaternary logic encoder-decoder circuit design using CMOS, *Proc. ISMVL*, 190-195.
- [26] **Sasao T.**, 1986. On the optimal design of multiple-valued PLA's, *Proc. IEEE 16th ISMVL*, 214-222.

- [27] **Jain A., Bolton R. and Abd-El-Barr M.**, 1993. CMOS Multiple-valued Logic Design-Part II; Function Realization, IEEE Trans. on Circuits and Systems, **40 no.8**, 515-522.
- [28] **Su S.Y.H. and Sarris A.A.**, 1972. The relationship between multivalued switching algebra and Boolean algebra under different definitions of complement, IEEE Trans. on Computers, **C-21**, 479-485.
- [29] **Vranesic Z.G., Lee E.S. and Smith K.C.**, 1970. A many-valued algebra for switching systems, IEEE Trans. on Computers, **C-19**, 964-971.
- [30] **Zilic Z. And Vranesic Z.G.**, 1995. A Multiple-Valued Reed-Muller Transform for Incompletely Specified Functions, IEEE Trans. on Computers, **44 no. 8**, 1012-1020.
- [31] **Temel T. and Morgül A.**, Impelementation of multi-valued logic, simultaneous literal operations with full CMOS current-mode threshold circuits, Electronic Letters, February 2002, **38 no.4**, 160-161.
- [32] **Davio M. and Deschamps J.P.**, 1981. Synthesis of discrete functions using I2L technology, IEEE Trans. on Computers, **C-30**, 653-661.
- [33] **Jain, A.K.; Abd-El-Barr, M.H.; Bolton, R.J.**, 1995, Current-mode CMOS multiple-valued logic function realization using a direct cover algorithm, Proceedings., 25th International Symposium on Multiple Valued Logic, 216 -221.
- [34] **Davio M., Deschamps J.P and Thayse A.**, 1978, Discrete and Switching Functions, McGraw-Hill, Massachusetts.
- [35] **Zilic Z. and Vranesic Z.**, 1993, Current-Mode CMOS Galois Field Circuits, Proceedings., 23th International Symposium on Multiple Valued Logic, 245-250
- [36] **McCluskey E.J**, 1979, Logic Design of multivalued I2L logic circuits, IEEE Trans. Computers, **C-28**, 546-559.
- [37] **Su S.Y.H. and Cheung P.T.**, 1972. Computer Minimization of Multivalued Switching Functions, IEEE Trans. on Computers, **C-21 no.9**, 995-1003.
- [38] **Besslich P.W.**, 1983. A method for the generation and processing of dyadic indexed data, IEEE Trans. on Computers, **C-32**, 487-494.
- [39] **Besslich P.W and Pichlbauer P.**, 1981. A fast transform procedure for the generation of near-minimal covers of Boolean functions, Proc. IEE, **128 Part E**, 250-254.

- [40] **Tirumalai P.P. and Butler J.T.**, Feb. 1991. Minimization algorithms for multiple-valued programmable logic arrays, *IEEE. Trans. on Computers*, C-40, 167-177.
- [41] **Dueck G.W.**, 1992, Direct cover MVL minimization with cost-tables, *Proceedings., Twenty-Second International Symposium on Multiple-Valued Logic*, 58 -65.
- [42] **Yurchak J.M. and Butler J.T.**, May 1990. HAMLET- an expression compiler/optimizer for the implementation of the heuristics to minimize multiple-valued programmable logic arrays, *Proc. ISMVL*, 144-152.
- [43] **Onneweer S., Kerkhoff H.G and Butler J.T.**, May 1988. Structural computer-aided design of current-mode CMOS logic circuits, *Proc. of ISMVL*, 21-30.
- [44] **Chang Y.B. and Butler J.T.**, May 1991. The design of Current Mode CMOS Multiple-Valued Circuits, *Proc. of 21st ISMVL*, 130-138.
- [45] **Abd-El-Barr, M.H.; Bolton, R.J.; Jain, A.K.**, 1993, Current-mode CMOS realization of a multiple-valued logic neurode, *WESCANEX 93. 'Communications, Computers and Power in the Modern Environment.'* Conference Proceedings., *IEEE* , 201 -207.
- [46] **Abd-El-Barr, M.; Hasan, M.N**, 1996, New MVL-PLA structures based on current-mode CMOS technology, *Proceedings., 26th International Symposium on Multiple-Valued Logic*, 98 -103.
- [47] **Hnayu T., Mochizuki A. and Kameyama M.**, Dec. 1996. Design and evaluation of a multiple-valued arithmetic integrated circuit based on differential logic, *IEE Proc. on Circuit Devices Systems*, 143 no. 6, 331-336.
- [48] **Pelayo F.J, Prieto A., Lloris A. and Ortega J.**, 1991. CMOS Current-Mode Multivalued PLA's, *IEEE Trans. on Circuits and Systems*, 38 no 4, 434-441.
- [49] **Jain A.K., Bolton R.J. and Abd-El Barr M.H.**, August 1993. CMOS Multiple-Valued Logic Design-Part I; Circuit Implementation, *Trans. on Circuits and Systems*, 40 no. 8, 503-514.
- [50] **Jain, A.K.; Bolton, R.J.; Abd-El-Barr, M.H.; Cheung, C .**,1993, On multiple-valued logic design of neural networks, *Proceedings of the 36th Midwest Symposium on Circuits and Systems* , 2, 1489 - 1492.
- [51] **Abo A. and Mehta S.**, Cmos Current Mode Adders, kabuki.eecs.berkeley.edu/~abo/papers/241/241_report.pdf

- [52] **Kuntman H.H.**, 1997, Analog MOS Tmdevre Teknięi, İT rektrlę 1587, İstanbul, 2.3-2.5.
- [53] **Meisel W.S.**, July 1968. Variable-Threshold Threshold Elements, IEEE Trans. on Computers, C-17 no. 7, 656-667.
- [54] **Kuntman H.H, Toker A., zcan S.**, 2000, Sayısal Elektronik Devreleri, İstanbul, 136-145.
- [55] **Onneweer S. and Kerkhoff H.**, May 1987. High radix current-mode CMOS c ircuit s b ased o n t he t runcated-difference o perator, P roc. 17th ISMVL, 188-195.
- [56] **Temel T. ve Morgl A.**, 2001, Implementation of multi-valued logic gates using full current-mode CMOS circuits, ELECO, Bursa, 53-57.
- [57] **Temel T.**, 2002, Current-mode CMOS design of multi-valued logic circuits, *Doktora Tezi*, B.. İstanbul.



EK A

Devre simülasyonlarında kullanılan transistor parametreleri:

UMC 0.18um LOGIC G-II technology, 1.8V PMOS Model

.subckt pmos D G S B

+ params: wp=29.6u lp=0.19u

MP1 D G S B psubckt l={lp} w={wp} ad={0.49u*wp} as={0.49u*wp}
+pd={0.49u+wp} ps={0.49u+wp} nrd={0.15u/wp} nrs={0.15u/wp}

.model psubckt pmos level=7 lmin=1.8e-7 lmax=5e-5 wmin=2.4e-7 wmax=1.0e-4
+tnom=27.0 version=3.1 tox=4.20e-09 xj=1.8000000e-07 nch=4.2198000e+17
+lln=0.3505198 lwn=1.0000000 wln=1.0000000 wwn=1.0541868 lint=-6.9385000e-
+08 ll=3.4482780e-10 lw=-1.0000000e-14 lwl=1.6088899e-17 wint=-3.7410000e-
+08 wl=2.0000000e-15 ww=-2.8621201e-15 ww1=5.0000000e-22 mobmod=1
+binunit=2 xl=-5.500e-09 xw=0.00 dwg=-2.7302200e-08 dwb=6.2500000e-09
+vth0=-0.4325310 k1=0.4752240 k2=3.7840110e-02 k3=0.1285200
+dvt0=1.2496901 dvt1=0.8907000 dvt2=-1.1200000e-03 dvt0w=1.5734400
+dvt1w=2.1000000e+06 dvt2w=0.4000000 nlx=4.1141920e-08 w0=2.4624358e-09
+k3b=2.5609901 vsat=1.8876778e+05 ua=-8.6959600e-10 ub=2.0968847e-18 uc=-
+3.0418199e-11 rdsw=1.4006900e+02 prwb=0 prwg=0.6013760 wr=1.0315588
+u0=7.1460100e-03 a0=1.8390000 keta=-1.1100899e-03 a1=0.00 a2=0.4000000
+ags=0.3401130 b0=4.3449200e-08 b1=1.4000000e-08 voff=-8.7928700e-02
+nfactor=1.4625300 cit=-7.9328180e-05 cdsc=1.6465606e-03 cdsb=4.1442100e-04
+cdscd=0.00 eta0=9.1642000e-02 etab=-4.8870300e-02 dsub=0.5212099
+pclm=1.1390049 pdiblc1=1.5900000e-03 pdiblc2=1.7381144e-04
+pdiblc3=9.4574010e-02 drou=0.5600000 psce1=4.5604730e+08
+psce2=9.0845550e-04 pvag=1.7533890 delta=1.4999999e-02 alpha0=0.00
+alpha1=0.00 beta0=30.0000000 kt1=-0.3800000 kt2=-5.2000000e-02
+at=1.0000000e+03 ute=-1.7500000 ual=-2.1000000e-10 ub1=-1.0100000e-18
+uc1=8.6000000e-11 kt11=-8.0000000e-09 prt=-4.0000000e+02 cj=0.00114
+mj=0.395 pb=0.762 cjsw=1.74e-10 mjsw=0.324 pbsw=0.665 cta=0.001
+ctp=0.000753 pta=0.00155 ptp=0.00124 js=1.30e-05 jsw=1.50e-11 cgdo=1.58e-10
+cgso=1.58e-10 cgbo=0 capmod=2 cjswg=3.78e-10 tlevc=1 nqsmod=0 xpart=1
+cf=1.533e-10 hdif=2.6e-07 rsh=8

.ends

UMC 0.18um LOGIC G-II technology, 1.8V nMOS Model

.subckt nmos D G S B

+ params: wn=29.6u ln=0.19u

MN1 D G S B nsubckt l={ln} w={wn} ad={0.49u*wn} as={0.49u*wn}
+pd={0.49u+wn} ps={0.49u+wn} nrd={0.15u/wn} nrs={0.15u/wn}

.model nsubckt nmos level=7 lmin=1.8e-7 lmax=5.0e-5 wmin=2.4e-7 wmax=1.0e-4
+tnom=27.0 version=3.1 tox=4.20000e-09 xj=1.6000000e-07 nch=3.5604000e+17
+lln=0.4744422 lwn=1.0000000 w ln=1.0000000 w wn=1.0395309 lint=3.240610e-
+08 ll=-1.9233681e-11 lw=-5.0000000e-21 lwl=1.4275000e-18 wint=1.9980000e-
+08 wl=-5.0000000e-21 ww=-2.1204030e-15 ww1=-2.4150000e-22 mobmod=1
+dwg=-5.3200000e-09 dwb=7.7019000e-09 vth0=0.3155591 k1=0.4181468 k2=-
+2.5054600e-02 k3=-0.8650800 dvt0=1.1079000 dvt1=0.4816967 dvt2=-
+4.0100000e-02 dvt0w=0.1961720 d vt1w=3.6750000e+05 d vt2w=-8.0665500e-02
+nlx=2.7241546e-07 w0=1.0000000e-20 k3b=1.1550000 vsat=1.0137400e+05 ua=-
+1.1684023e-09 ub=2.5709233e-18 uc=8.3571000e-11 rdsw=72.9541100 prwb=-
+0.1340100 prwg=0.2000000 wr=0.9442700 u0=3.2613610e-02 a0=1.9627345
+keta=-1.1009000e-03 a1=0.00 a2=0.9900000 ags=0.5447900 b0=4.8668600e-08
+b1=1.4053400e-08 voff=-0.0942040 nfactor=1.1893200 cit=-2.5854103e-04
+cdsc=2.4786401e-03 cdsb=1.2005900e-04 cdsd=0.00 eta0=7.1668900e-02
+etab=-1.7981000e-03 dsub=0.3790412 pclm=8.5993000 pdiblc1=5.3830910e-03
+pdiblc2=2.0840220e-04 pdiblc3=0.1714839 drou=0.8400000
+pscb1=6.3000000e+08 pscbe2=1.0000000e-20 pvag=4.0709360
+delta=1.0000000e-03 alpha0=0.00 beta0=30.0000000 kt1=-0.2980000 kt2=-
+5.2000000e-02 at=5.2000000e+04 ute=-1.7900000 ual=1.8300000e-09 ub1=-
+2.6100001e-18 uc1=1.2000000e-10 kt11=-5.0000000e-09 prt=-3.1000000e+02
+cj=0.00119 mj=0.515 pb=0.79 cjsw=1.60e-10 mjsw=0.381 pbsw=0.705
+cta=0.000919 ctp=0.000914 pta=0.00158 ptp=0.000924 js=3.39e-05 jsw=3.26e-11
+cgdo=1.21e-10 cgso=1.21e-10 cgbo=0 capmod=2 cjswg=2.02e-10 tlevc=1
+nqsmod=0 xpart=1 cf=1.533e-10 hdif=2.6e-07 rsh=8

.ends

ÖZGEÇMİŞ

1975 yılında Bulgaristan Silistrede doğdu. İlk öğrenimini Bulgaristan'da tamamladı. 1989 yılında Türkiye'ye göç etti. Lise öğrenimini Suphi Koyuncuoğlu Lisesi/İzmir'de tamamladı. 1996 yılında İTÜ Elektrik-Elektronik Fakültesi Elektrik Müh. bölümüne kabul edildi. 2000 yılında mühendislik eğitimini tamamladıktan sonra İTÜ Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği bölümünde yüksek lisans eğitimine başladı. Halen Yüksek Lisans tez çalışmasını sürdürmektedir.

