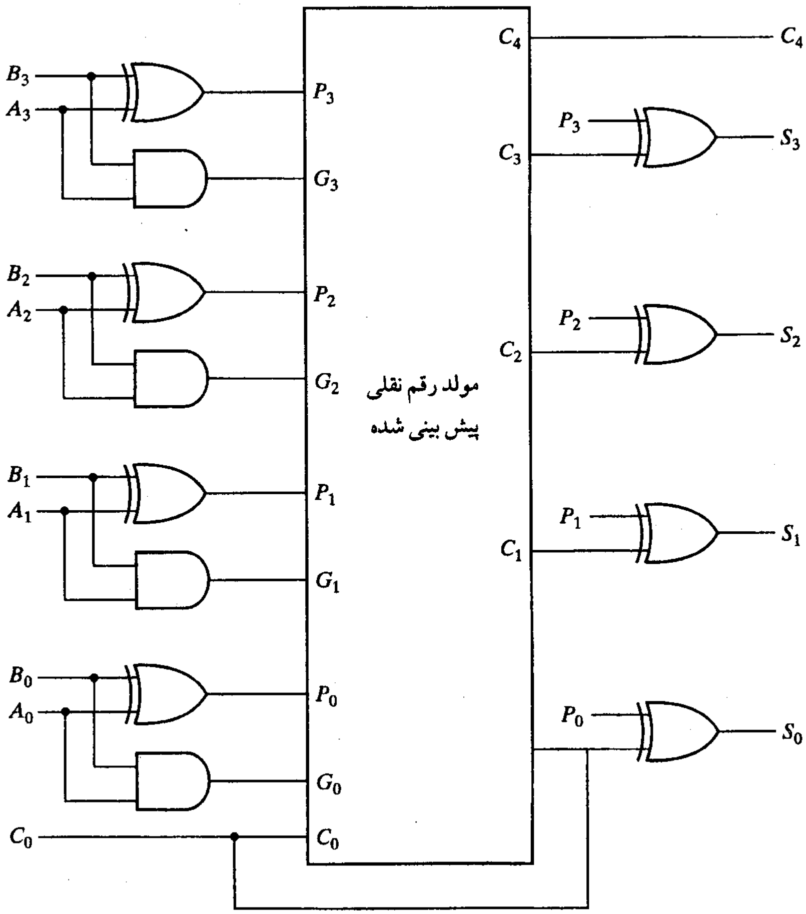
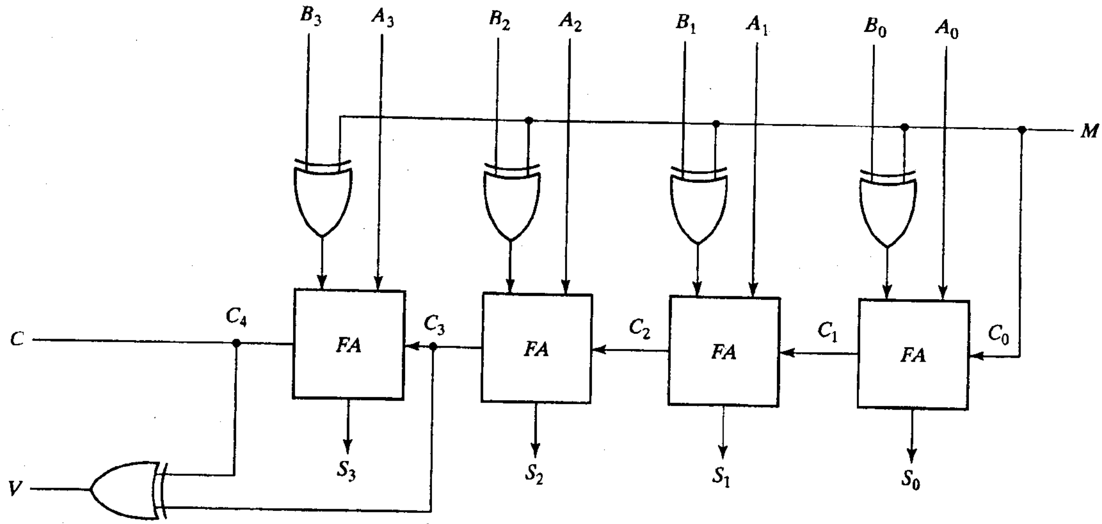


شکل ۱۲-۴. جمع‌کننده 4 بیت با پیش‌بینی نقلی

تفریق دودویی

در بخش ۵-۱ دیدیم که تفریق اعداد دودویی بی‌علامت با استفاده از متمم راحت‌تر انجام می‌گیرد. به خاطر دارید که A−B را می‌توان با محاسبه متمم 2 عدد B و جمع آن با A معین کرد. متمم 2 را با بدست آوردن متمم 1 و جمع آن با 1 محاسبه می‌کنند. متمم 1 را هم با وارونگر بدست آورده و عدد 1 را هم از طریق ورودی نقلی به آن اضافه می‌نمایند.

مدار تفریق‌گر A−B متشکل از یک جمع‌کننده با وارونگرهای واقع در بین ورودی B و ورودی مربوطه‌اش در تمام جمع‌کننده می‌باشد. نقلی ورودی C_0 به هنگام تفریق باید برابر با 1 شود. بنابراین عمل به صورت A بعلاوه متمم 1 عدد B بعلاوه 1 اجرا می‌شود. این عمل برابر با جمع A با متمم 2 عدد B خواهد بود. برای اعداد بی‌علامت اگر A≥B باشد، عمل فوق A−B و اگر A<B باشد (B−A) است. برای اعداد علامت‌دار، نتیجه A−B است به شرطی که سرریز وجود نداشته باشد (بخش ۶-۱ ملاحظه شود).



شکل ۱۳-۴ . جمع- تفریق‌گر

عملیات جمع و تفریق را می‌توان با یک مدار در هم ادغام کرده و با یک جمع‌کننده دودویی مشترک انجام داد. این کار با افزودن یک گیت XOR در هر جمع‌کننده کامل صورت می‌گیرد. یک مدار جمع- تفریق‌گر در شکل ۱۳-۴ دیده می‌شود. وقتی $M = 0$ است، مدار یک جمع‌کننده و وقتی $M = 1$ باشد، مدار یک تفریق‌گر خواهد بود. هر گیت XOR ورودی M و یکی از ورودی‌های B را دریافت می‌کند. وقتی $M = 0$ است داریم، $B \oplus 0 = B$. جمع‌کننده کامل B را دریافت می‌نماید، نقلی ورودی 0 است و بنابراین مدار عمل A بعلاوه B را اجرا می‌کند. اگر $M = 1$ باشد، $B \oplus 1 = B'$ بوده و $C_0 = 1$ است. ورودی‌های B همگی متمم شده و از طریق ورودی نقلی، یک 1 به آن اضافه می‌شود. در این حالت مدار یک عمل A بعلاوه متمم 2 عدد B را انجام می‌دهد. (XOR با خروجی V ، یک سرریز را شناسایی می‌نماید).

لازم است متذکر شویم که در سیستم متمم علامت‌دار، اعداد دودویی هم چون اعداد بی‌علامت، با قوانین جمع و تفریق یکسانی ترکیب می‌شوند. بنابراین، کامپیوترها نیاز به یک سخت‌افزار مشترک دارند تا هر دو نوع محاسبه را انجام دهند. کاربر یا برنامه‌نویس باید نتایج چنین جمع یا تفریقی را متفاوت تفسیر کنند و این به علامت‌دار یا بی‌علامت بودن اعداد بستگی دارد.

سرریز

هرگاه دو عدد n رقمی با هم جمع شوند و حاصل جمع n+1 رقم را اشغال کند، گوییم سرریز رخ داده است. این مطلب جدا از علامت‌دار بودن یا نبودن برای اعداد دهدهی یا دودویی صحیح است. وقتی که جمع با کاغذ و قلم انجام می‌شود، سرریز مسئله‌ای نیست زیرا محدودیتی برای عرض صفحه جهت نوشتن جمع وجود ندارد. ولی سرریز در کامپیوترهای دیجیتال مشکلاتی ایجاد می‌کند، زیرا تعداد

بیت‌های نگهدارنده عدد محدود بوده و نتیجه‌ای را که n+1 بیت دارد نمی‌توانند در خود جای دهند. به این دلیل، بسیاری از کامپیوترها وقوع یک سرریز را، اگر رخ دهد، شناسایی می‌کنند و فیلیپ فلاپ مربوطه را در 1 می‌نشانند تا بعد به وسیله کاربر چک شود.

تشخیص یک سرریز پس از جمع دو عدد دودویی به این بستگی دارد که آیا اعداد علامت دارند یا بی‌علامت‌اند. وقتی دو عدد بی‌علامت با هم جمع شوند، یک سرریز از نقلی باارزش‌ترین مکان تشخیص داده می‌شود. در حالتی که اعداد علامت‌دار باشد، سمت چپ‌ترین بیت همواره علامت را نشان داده و اعداد منفی هم به صورت متمم 2 هستند. وقتی دو عدد علامت‌دار جمع شوند، با بیت علامت به عنوان بخشی از عدد رفتار می‌شود و رقم نقلی انتهایی هیچ سرریزی را مشخص نمی‌کند.

در جمع وقتی که یکی از اعداد مثبت و دیگری منفی باشد، سرریز رخ نمی‌دهد، زیرا جمع یک عدد مثبت با یک عدد منفی نتیجه‌ای تولید می‌کند که از بزرگترین آن دو کوچکتر است. سرریز هنگامی رخ می‌دهد که هر دو عدد جمع شونده مثبت یا منفی باشند. برای درک بهتر موضوع مثال زیر را ملاحظه کنید. دو عدد علامت‌دار دودویی 70+ و 80+ دودویی در دو ثبات 8 بیتی ذخیره شده‌اند. محدوده اعدادی که هر یک از ثبات‌ها داراست از 127+ تا 128− دودویی است. چون مجموع دو عدد 150+ است، حاصل از ظرفیت ثبات 8 بیتی تجاوز خواهد کرد. این مطالب هنگامی که هر دو عدد مثبت یا منفی باشند صحت دارد. دو جمع مذکور همراه با ارقام نقلی در زیر نشان داده شده‌اند:

نقلی‌ها :	1	0
−70	1	0111010
−80	1	0110000
−150	0	1101010

نقلی‌ها :	0	1
+70	0	1000110
+80	0	1010000
+150	1	0010110

توجه کنید که حاصل جمع هشت بیتی که باید مثبت باشد یک بیت علامت منفی دارد و نتیجه 8 بیتی که باید منفی باشد دارای بیت علامت مثبت است. با این وجود اگر رقم نقلی خارج شده از بیت علامت به عنوان بیت علامت در نظر گرفته شود، آنگاه جواب 9 بیتی حاصل صحیح خواهد بود. چون پاسخ نمی‌تواند در 8 بیت جای داده شود، گوییم سرریز رخ داده است.

وضعیت سرریز را می‌توان با وجود رقم نقلی به بیت علامت و نقلی خروجی از بیت علامت مشاهده کرد. اگر این دو نقلی یکی نباشند، یک سرریز رخ داده است. این نکته در مثال‌های فوق که در آن دو نقلی به طور جداگانه نشان داده شده‌اند دیده می‌شود. اگر دو رقم نقلی را به یک گیت XOR اعمال کنیم، وقوع سرریز با 1 شدن خروجی این گیت شناسایی می‌شود. برای این که روش به خوبی کار کند متمم 2 باید از طریق بدست آوردن متمم 1 و جمع آن با 1 انجام گردد. این کار موجب مراقبت از حالتی می‌شود که در آن عدد منفی ماکزیمم متمم شود.

مدار جمع- تفریق‌گر با خروجی‌های C و V در شکل ۱۳-۴ دیده می‌شود. اگر دو عدد دودویی بی‌علامت تصور شوند، آنگاه بیت C ، نقلی بعد از جمع یا قرض بعد از تفریق است. اگر اعداد علامت‌دار

فرض شوند، آنگاه بیت V یک سرریز را مشخص می‌کند. اگر $V = 0$ بعد از یک جمع یا تفریق باشد، بیانگر نبود سرریز بوده و نتیجه n بیتی حاصل صحیح است. اگر $V = 1$ باشد، در این صورت نتیجه عمل حاوی $n+1$ بیت می‌باشد، ولی بیت $n+1$ام علامت واقعی است که به یک مکان بیرونی منتقل شده است.

۴-۵ جمع‌کننده دهدهی

کامپیوترها یا ماشین‌های حسابی که اعمال محاسباتی را مستقیماً در سیستم اعداد دهدهی انجام می‌دهند، اعداد دهدهی را به فرم کد دودویی ارائه می‌کنند. یک جمع‌کننده در این نوع کامپیوترها، از نوعی مدار محاسباتی استفاده می‌کند که اعداد دهدهی کد شده را می‌پذیرد و نتایج را در همان کد ارائه می‌نماید. برای جمع دودویی کافی است جفت بیت باارزش را همراه با رقم نقلی قبلی در نظر بگیرید. یک جمع‌کننده دهدهی به حداقل ده ورودی و پنج خروجی نیاز دارد زیرا برای کد هر رقم دهدهی چهار بیت لازم است و مدار باید ورودی و خروجی نقلی هم داشته باشد. برای انجام این‌گونه جمع، مدارهای جمع‌کننده دهدهی متعددی وجود دارند که انتخاب آنها به کد به‌کار رفته در نمایش ارقام دهدهی بستگی دارد. در اینجا ما جمع‌کننده دهدهی را برای کد BCD بررسی می‌کنیم.

جمع‌کننده BCD

جمع حسابی دو رقم دهدهی در BCD را همراه با یک رقم نقلی از مرحله قبل در نظر بگیرید. چون هر رقم ورودی از 9 تجاوز نمی‌کند، حاصل جمع خروجی از $9 + 9 + 1 = 19$ بیشتر نخواهد شد. عدد 1 در جمع فوق، نقلی ورودی است. فرض کنید که دو رقم BCD را به جمع‌کننده دودویی 4 بیتی اعمال نماییم. جمع‌کننده، حاصل جمع را به فرم دودویی اجرا می‌کند و نتیجهٔ تولید شده بین 0 تا 19 خواهد بود. این اعداد دودویی در جدول (۵-۴) مشاهده می‌شود که با K، z_8، z_4، z_2 و z_1 برچسب خورده‌اند. K یک رقم نقلی است و اندیس زیر حرف z وزن‌های 8، 4، 2 و 1 می‌باشند که به چهار بیت کد BCD تخصیص یافته‌اند. ستون زیر حاصل جمع دودویی، مقادیر دودویی ظاهر شده در خروجی‌های جمع‌کننده چهار بیت را نشان می‌دهد. حاصل جمع خروجی دو رقم دهدهی باید به فرم BCD درآید و نیز باید آن طور که در زیر ستون جمع BCD ملاحظه می‌شود ظاهر گردد. مسئله این است که برای تبدیل جمع دودویی به رقم BCD عدد که در ستون جمع BCD مشاهده می‌شود باید قانونی پیدا شود.

ضمن بررسی محتوای جدول، ملاحظه می‌شود که وقتی جمع دودویی برابر با یا کمتر از 1001 باشد، با عدد BCD نظیر خود برابر است، و بنابراین تبدیلی لازم نیست. وقتی جمع دودویی بزرگتر از 1001 باشد، نمایش بی‌اعتباری را برای BCD خواهیم داشت. افزایش دودویی 6 (0110) به جمع دودویی آن را به نمایش BCD صحیح تبدیل می‌کند، ضمن این که یک رقم نقلی نیز در صورت لزوم تولید خواهد کرد.

مدار منطقی برای تشخیص این اصلاح، می‌تواند از وارده‌های جدول حاصل گردد. واضح است که وقتی نقلی خروجی $K = 1$ باشد نیاز به اصلاح جمع دودویی وجود دارد. دیگر ترکیبات شش‌گانه از

جدول ۵-۴. طریقه طراحی جمع کننده BCD

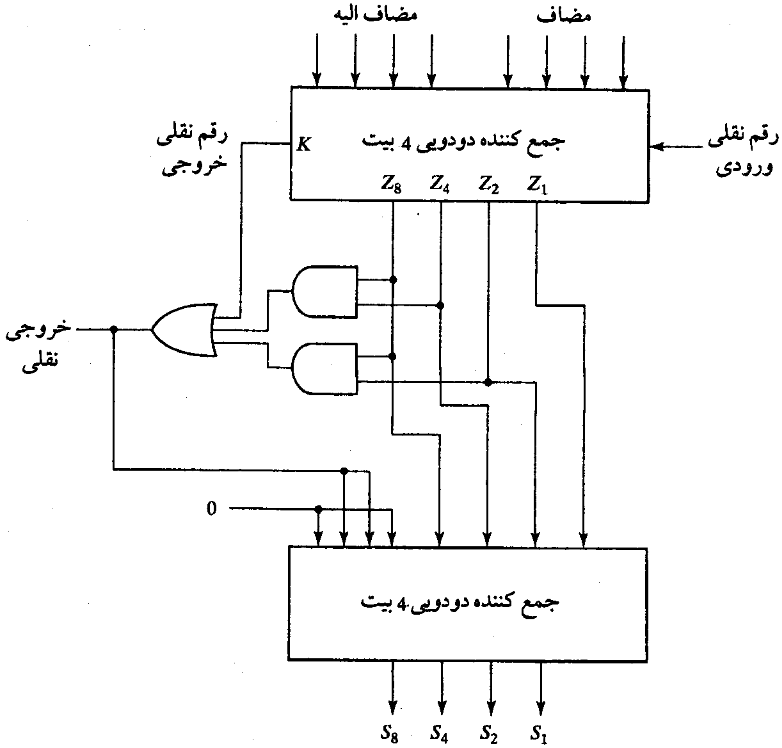
جمع دودویی					جمع BCD					دهدهی
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

1010 تا 1111 که به اصلاح نیاز دارند دارای 1 در مکان z_8 می‌باشند. برای تفکیک این شش حالت از 1000 و 1001، که آنها نیز دارای 1 در مکان z_8 هستند، به z_4 و z_2 مراجعه می‌کنیم که در هر حال حداقل یکی از آنها 1 است. به این ترتیب شرط اصلاح و داشتن یک نقلی خروجی را می‌توان با تابع بولی زیر بیان کرد:

$$C = K + Z_8Z_4 + Z_8Z_2$$

وقتی $C = 1$ است، لازم است 0110 به جمع دودویی اضافه شود تا یک نقلی خروجی برای طبقه بعدی فراهم شود.

یک جمع‌کننده BCD که دو رقم BCD را با هم جمع کرده و ارقام جمع را به BCD نشان می‌دهد در شکل ۱۴-۴ ملاحظه می‌گردد. دو رقم دهدهی همراه با نقلی ورودی ابتدا در جمع‌کننده 4 بیت فوقانی جمع شده و حاصل جمع دودویی تولید می‌کنند. وقتی نقلی خروجی برابر 0 باشد، چیزی به جمع دودویی اضافه نمی‌شود. وقتی این نقلی برابر 1 باشد، عدد دودویی 0110 از طریق جمع‌کننده 4 بیت پایینی به جمع دودویی اضافه می‌گردد. نقلی خروجی تولید شده در جمع‌کننده پایین می‌تواند صرف‌نظر شود زیرا اطلاعاتی را حمل می‌کند که قبلاً در پایانه نقلی خروجی وجود داشته است. یک جمع‌کننده دهدهی موازی که n رقم دهدهی را جمع می‌کند به n طبقه جمع‌کننده BCD نیاز دارد. نقلی خروجی هر طبقه باید به ورودی طبقه بالاتر متصل گردد.



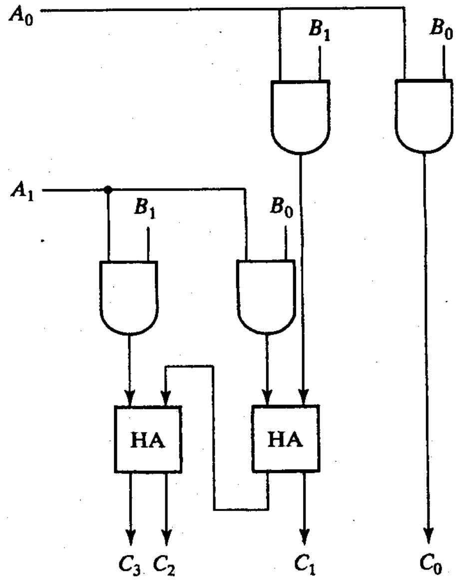
شکل ۱۴-۴. نمودار بلوکی یک جمع‌کننده BCD

۶-۴ ضرب دودویی

ضرب اعداد دودویی همچون ضرب اعداد دهدهی انجام می‌شود. هر بیت مضروب، در کم‌ارزش‌ترین بیت مضروب فیه ضرب می‌شود. چنین حاصلضربی، حاصلضرب جزیی خوانده می‌شود. حاصلضرب‌های جزیی هر بار یک مکان به چپ انتقال می‌یابند. حاصلضرب نهایی از جمع حاصلضرب‌های جزیی بدست می‌آید.

برای این که ببینیم که یک ضرب کننده چگونه با یک مدار ترکیبی پیاده می‌شود، ضرب اعداد دو بیت را طبق شکل ۱۵-۴ در نظر بگیرید. بیت‌های مضروب، B_0 و B_1 و بیت‌های مضروب‌فیه A_0 و A_1 و حاصلضرب $C_3C_2C_1C_0$ فرض می‌شوند. اولین حاصلضرب جزیی با ضرب A_0 در B_1B_0 حاصل می‌گردد. ضرب دو بیت مثل A_0 و B_0 هنگامی 1 تولید می‌کند که هر دوی آنها 1 باشند؛ در غیر این صورت 0 تولید خواهد کرد. این پاسخ مشابه با عمل AND است. بنابراین حاصل ضرب جزیی را می‌توان با گیت‌های AND مطابق شکل پیاده کرد. دومین حاصلضرب جزیی از ضرب A_1 در B_1B_0 بدست می‌آید که باید یک مکان هم به چپ جابجا شود. دو حاصلضرب جزیی به وسیله مدار دو نیم جمع‌کننده (HA) با هم جمع می‌شوند.

مضروب			B_1	B_0
مضروب فیه			A_1	A_0
			A_0B_1	A_0B_0
		A_1B_1	A_1B_0	
	C_3	C_2	C_1	C_0



شکل ۱۵-۴. ضرب دودویی 2 بیت در 2 بیت

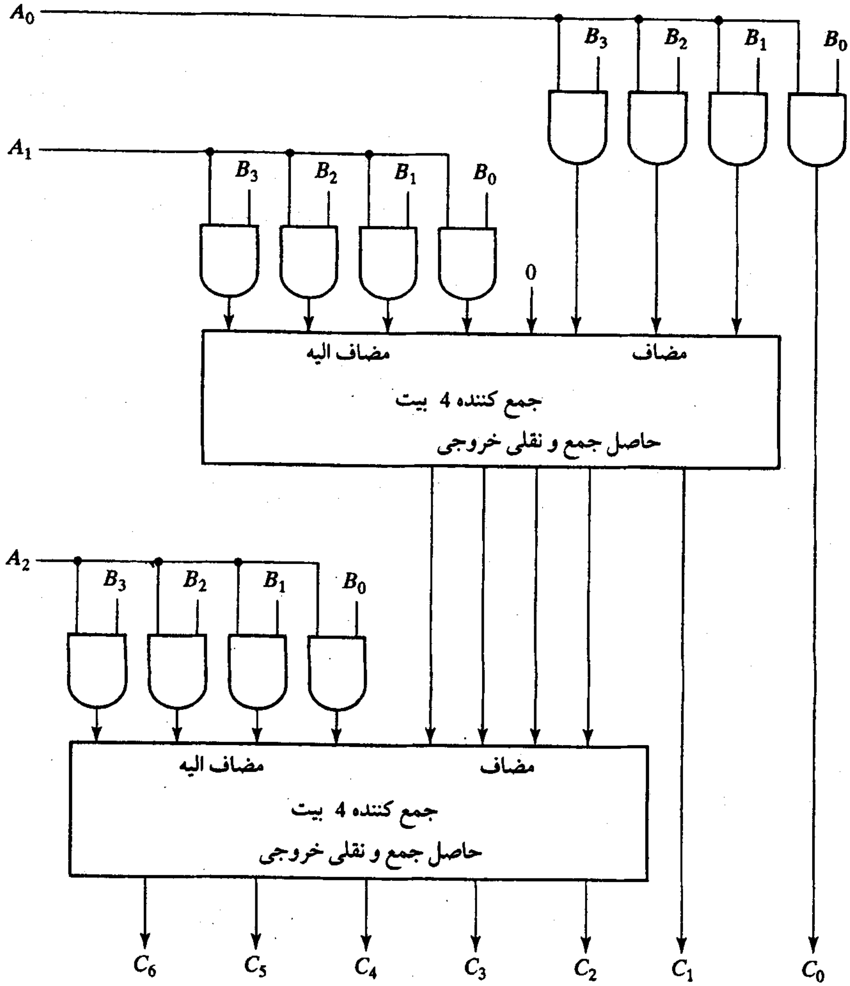
معمولاً در حاصلضرب‌های جزیی بیت‌های بیشتری وجود دارند و لازم است از تمام جمع‌کننده برای تولید جمع حاصلضرب‌های جزیی استفاده شود. توجه کنید لزومی ندارد که کم‌ارزش‌ترین بیت از جمع‌کننده عبور کند زیرا با خروجی اولین گیت AND ، تشکیل شده است.

به طریقی مشابه می‌توان یک مدار ترکیبی ضرب دودویی با بیت‌های بیشتر ساخت. هر بیت از مضروب‌فیه در بیت‌های مضروب، AND می‌گردد. خروجی دودویی در هر سطحی از گیت‌های AND با حاصلضرب جزیی سطح قبلی برای تشکیل حاصلضرب جزیی جدید جمع می‌شود. آخرین سطح حاصلضرب کل را تولید می‌کند. برای J بیت مضروب فیه و K بیت مضروب به $(J \times K)$ گیت AND و $(J-1)$ عدد جمع‌کننده K بیت نیاز است تا حاصلضرب $J + K$ بیتی تولید شود.

به عنوان دومین مثال مدار ضرب کننده‌ای را ملاحظه نمایید که یک عدد دودویی 4 بیتی را در یک عدد 3 بیتی ضرب می‌کند. فرض کنید مضروب با $B_3B_2B_1B_0$ و مضروب‌فیه $A_2A_1A_0$ باشد. چون $K = 4$ و $J = 3$ است 12 گیت AND و دو جمع‌کننده 4 بیت برای تولید حاصلضرب 7 بیتی لازم است. نمودار منطقی ضرب کننده در شکل ۱۶-۴ دیده می‌شود.

۷-۴ مقایسه‌گر مقدار

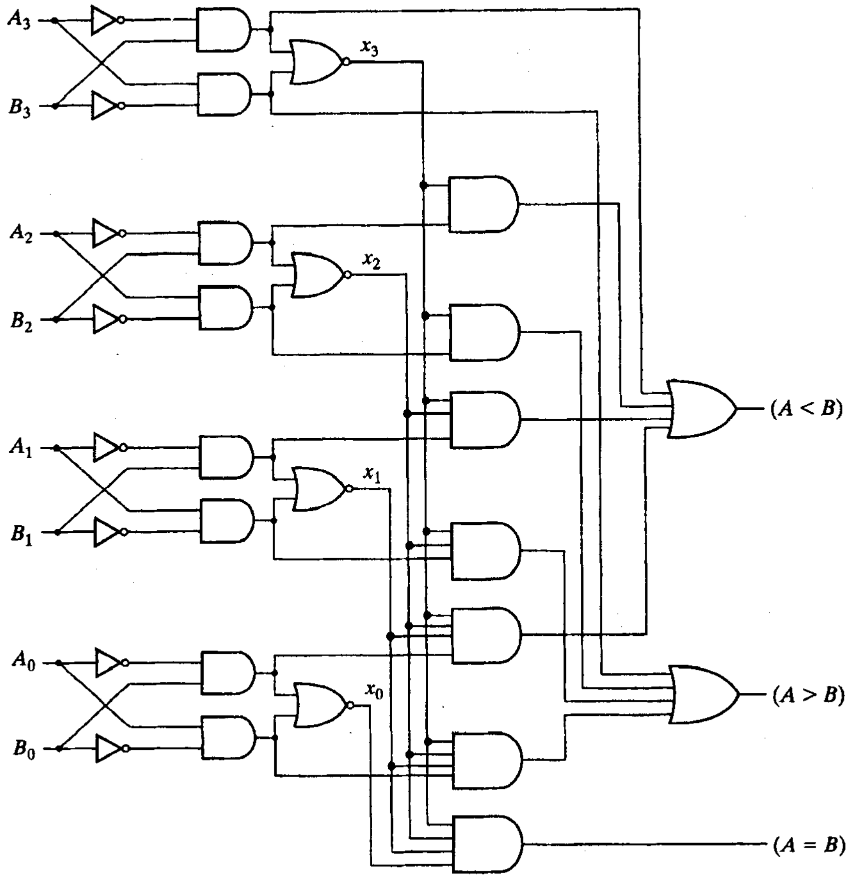
مقایسه دو عدد عملی است که توسط آن بزرگتر بودن، کوچکتر بودن یا مساوی بودن آنها معین می‌شود. یک مقایسه گر مقدار مداری ترکیبی است که دو عدد A و B را مقایسه می‌نماید و اندازه نسبی آنها را تعیین می‌کند. نتیجه این مقایسه با سه متغیر دودویی که بیانگر $A > B$ یا $A < B$ می‌باشد، مشخص می‌گردد.



شکل ۱۶-۴. ضرب کننده دودویی 4 بیت در 3 بیت

مدار مقایسه دو عدد n بیت 2^{2n} وارده در جدول دارد و حتی با n=3 خسته کننده خواهد شد. از طرف دیگر، یک مدار مقایسه‌گر ممکن است مقدار قابل توجهی نظم در خود داشته باشد. توابع دیجیتال که ذاتاً دارای نظمی درونی هستند، معمولاً با روال‌های الگوریتمی قابل طراحی‌اند. یک الگوریتم روالی است که مجموعه مراحل معینی را مشخص می‌نماید و اگر دنبال شوند، از آن حلی برای مسئله حاصل می‌گردد. ما در اینجا از این روش برای ارائه یک الگوریتم جهت پیاده‌سازی مقایسه‌گر مقدار 4 بیتی استفاده خواهیم کرد.

در اینجا الگوریتم، کاربرد مستقیم روالی است که فرد برای مقایسه نسبی اندازه‌های دو عدد به کار می‌برد. دو عدد A و B را که هر کدام چهار رقم دارند در نظر بگیرید. ضرایب اعداد را به ترتیب نزولی زیر



شکل ۱۷-۴. مقایسه گرمقدرا چهار بیتی

هم می‌نویسیم:

$$A = A_3A_2A_1A_0$$
$$B = B_3B_2B_1B_0$$

هر حرف اندیس‌دار یک رقم را در عدد نشان می‌دهد. دو عدد هنگامی مساوی‌اند که همه جفت ارقام متناظر با هم برابر باشند: یعنی $A_3 = B_3$، $A_2 = B_2$، $A_1 = B_1$ و $A_0 = B_0$. وقتی که اعداد دودویی باشند، ارقام 1 یا 0اند و رابطه تساوی هر جفت بیت به طور منطقی با یک تابع XOR نمایش داده می‌شود.

$$x_i = A_iB_i + A_i'B_i' \qquad \text{for } i = 0, 1, 2, 3$$

که در آن $x_i = 1$ به شرطی صحت دارد که بیت‌های مکان iام برابر باشند (یعنی اگر هر دو 1 یا هر دو 0).

برابری دو عدد A و B در مدار ترکیبی با یک متغیر خروجی و با علامت $(A = B)$ نشان داده می‌شود. این متغیر دودویی هنگامی 1 است که همه اعداد ورودی A و B مساوی باشند، در غیر این صورت 0 است. برای این که شرایط برابری برقرار باشد همه متغیرهای x_i باید برابر 1 شوند. در این

صورت AND همه متغیرها دیکته خواهد شد:

$$(A = B) = x_3x_2x_1x_0$$

عدد دودویی (A = B) هنگامی 1 است که فقط همه جفت ارقام دو عدد برابر باشند.

برای این که معین کنیم آیا A بزرگتر یا کوچکتر از B است، اندازه‌های نسبی دو رقم را با شروع از باارزش‌ترین مکان آغاز می‌نماییم. اگر دو رقم مساوی باشند، دو رقم پایین‌تر را مقایسه می‌کنیم. این مقایسه تا رسیدن به یک جفت غیرمساوی ادامه خواهد داشت. اگر در این هنگام بیت متعلق به A برابر 1 و B برابر 0 باشد، نتیجه می‌گیریم A > B است. اگر برعکس رقم مربوط به A برابر 0 و B برابر با 1 باشد، خواهیم داشت A < B. مقایسه فوق را می‌توان با کمک دو تابع بولی به صورت زیر نوشت:

$$(A > B) = A_3B_3' + x_3A_2B_2' + x_3x_2A_1B_1' + x_3x_2x_1A_0B_0'$$
$$(A < B) = A_3'B_3 + x_3A_2'B_2 + x_3x_2A_1'B_1 + x_3x_2x_1A_0'B_0$$

سمبل‌های (A > B) و (A < B) متغیرهای خروجی دودویی هستند که بترتیب هنگام A > B یا A<B برابر 1 می‌شوند.

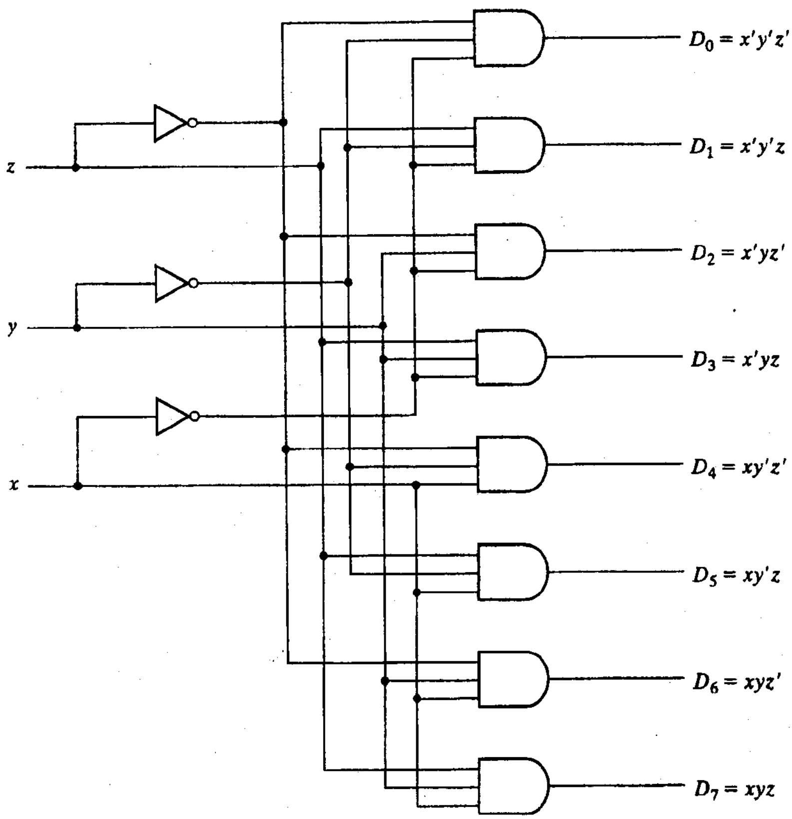
پیاده‌سازی گیتی سه متغیر خروجی ساده‌تر از آنچه به نظر می‌رسد انجام می‌شود زیرا شامل مقدار قابل توجهی اعمال تکراری است. خروجی‌های نامساوی می‌توانند از گیت‌هایی که برای تولید خروجی مساوی لازم بود استفاده کنند. نمودار منطقی یک مقایسه‌گر مقدار 4 بیتی در شکل ۱۷-۴ ملاحظه می‌شود. چهار خروجی x با مدارهای XNOR تولید شده و به گیت AND اعمال شده‌اند تا متغیر دودویی خروجی (A = B) تولید گردد. دو خروجی دیگر از متغیر x برای تولید توابع بولی لیست شده قبلی استفاده می‌کنند. این یک پیاده‌سازی چند طبقه است که الگوی منظمی دارد. روال برای بدست آوردن مدارهای مقایسه‌گر اندازه برای اعداد دودویی با بیش از چهار بیت از این مثال کاملاً آشکار است.

۸-۴ دیکدرها

کمیت‌های گسسته اطلاعاتی در سیستم‌های دیجیتال با کدهای دودویی نشان داده می‌شوند. یک کد دودویی n بیتی قادر است تا 2^n عنصر گسسته اطلاعات کد شده را نشان دهد. یک دیکدر مداری ترکیبی است که اطلاعات دودویی را از n خط ورودی به حداکثر 2^n خط خروجی منحصر به فرد تبدیل می‌کند اگر کد n بیتی دارای ترکیبات بی‌استفاده باشد، دیکدر ممکن است خروجی‌های کمتر از 2^n داشته باشد.

دیکدرهایی که در اینجا ارائه شده‌اند دیکدرهای n به m خوانده می‌شوند که $m \leq 2^n$ است. هدف از آنها تولید 2^n مینترم (یا کمتر) از n متغیر ورودی است. نام دیکدر همراه با دیگر مبدل‌های کد مانند دیکدر BCD به هفت قسمتی هم به کار می‌رود.

به عنوان مثال دیکدر 3 به 8 قسمتی شکل ۱۸-۴ را ملاحظه نمایید. سه ورودی به هشت خروجی دیکدر شده است که هر یک نمایشگر یکی از مینترم‌های متعلق به سه متغیر ورودی است. سه وارونگر، متمم ورودی‌ها را تهیه کرده و هشت گیت AND هر کدام یک مینترم تولید می‌کنند. کاربرد رایج این نوع دیکدر، تبدیل دودویی به هشت هشتی است. متغیرهای ورودی یک عدد دودویی را نشان می‌دهند، و



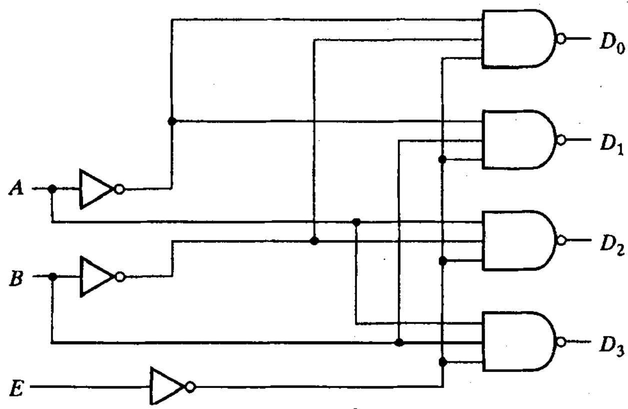
شکل ۱۸-۴. دیکدر 3 به 8

خروجی بیانگر هشت رقم در سیستم اعداد مبنای هشت است. با این وجود دیکدر 3 به 8 خط را می‌توان برای دیکد کردن هر کد 3 بیت در تولید هشت خروجی، یکی برای هر عنصر از کد، به کار برد.

طرز کار یک دیکدر می‌تواند با لیستی در جدول درستی ۶-۴ آشکار شود. برای هر ترکیب ورودی

جدول ۶-۴. جدول درستی دیکدر 3 به 8 خط

ورودی‌ها			خروجی ها							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



(الف) دیاگرام منطقی

E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

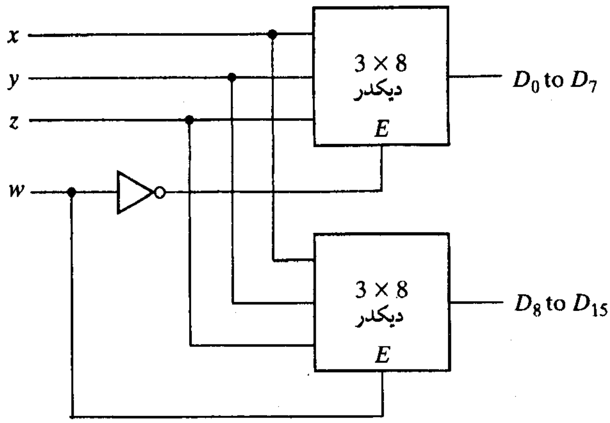
(ب) جدول درستی

شکل ۱۹-۴. دیکدر 2 به 4 خط با ورودی فعال ساز

ممکن، هفت خروجی وجود دارد که برابر 0 هستند و فقط یکی از آنها برابر 1 است. خروجی حاوی 1 بیانگر مینترم عدد دودیی حاضر در خطوط ورودی است.

بعضی از دیکدرها با گیت‌های NAND ساخته می‌شوند. چون گیت NAND عمل AND را با یک خروجی معکوس تولید می‌کند، تولید مینترم‌های دیکدر در شکل متمم اقتصادی‌تر است. به علاوه، دیکدر معمولاً دارای یک یا دو ورودی تواناساز یا فعال‌ساز برای کنترل‌کار مدار می‌باشند. یک دیکدر 2 به 4 با یک ورودی فعال‌ساز که با گیت‌های NAND ساخته شده در شکل ۱۹-۴ دیده می‌شود. مدار با خروجی‌های متمم شده و یک ورودی فعال‌ساز متمم شده کار می‌کند. دیکدر هنگامی که E برابر 0 باشد فعال می‌گردد. همانطور که توسط جدول درستی مشاهده می‌شود، هر بار تنها یک خروجی برابر 0 بوده و دیگر خروجی‌ها در وضعیت 1 قرار دارند. وقتی $E = 1$ باشد مدار غیرفعال است و به دو ورودی دیگر بستگی ندارد. هنگام غیرفعال شدن مدار، هیچ یک از خروجی‌ها در 0 نبوده و هیچ یک از مینترم‌ها انتخاب نمی‌شوند. به طور کلی، یک دیکدر ممکن است خروجی‌های متمم شده یا متمم نشده داشته باشد. ورودی فعال‌ساز ممکن است با سیگنال 0 یا 1 فعال گردد. بعضی از دیکدرها دارای دو یا چند ورودی فعال‌ساز می‌باشند که باید یک شرط منطقی مفروضی را برآورده سازند تا مدار فعال شود.

یک دیکدر با ورودی فعال‌ساز می‌تواند به عنوان یک دی‌مولتی پلکسر عمل کند. دی‌مولتی پلکسر مداری است که اطلاعات را از یک خط دریافت کرده و آن را به یکی از 2^n خط خروجی ممکن هدایت می‌نماید. انتخاب یک خروجی خاص با ترکیب بیتی n خط انتخاب صورت می‌گیرد. دیکدر شکل ۱۹-۴ را می‌توان به عنوان یک دی‌مولتی پلکسر 1 به 4 به کار برد. در این مدار E به عنوان ورودی داده و A و B ورودی‌های انتخاب هستند. تنها متغیر ورودی E مسیری به تمام چهارخروجی دارد، ولی اطلاعات ورودی تنها به یکی از خروجی‌ها هدایت می‌شود. این خروجی با ترکیب دودویی دو خط



شکل ۲۰-۴. دیکدر 4×16 ساخته شده با دو دیکدر 3×8

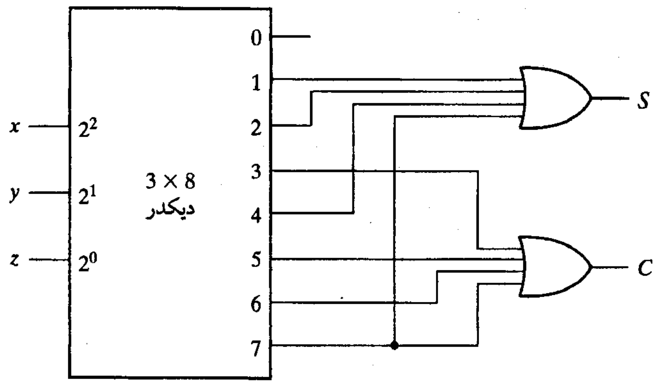
انتخاب A و B انتخاب می‌گردد. می‌توان انتخاب مسیر را از جدول درستی تحقیق کرد. مثلاً اگر خطوط انتخابی $AB = 10$ باشند، خروجی D_2 مثل ورودی E خواهد بود در حالی که دیگر خروجی‌ها در 1 نگهداشته خواهند شد. چون عمل دیکدر و دی‌مولتی پلکسر با استفاده از یک مدار حاصل می‌شود، یک دیکدر با ورودی فعال‌ساز را دیکدر/ دی‌مولتی پلکسر هم می‌خوانند.

برای تهیه مدار دیکدر بزرگتر می‌توان دیکدرها با ورودی‌های فعال‌ساز را به هم متصل کرد. شکل ۲۰-۴ دو دیکدر 3 به 8 را با ورودی‌های فعال‌ساز به هم پیوسته برای تشکیل یک دیکدر 4 به 16 خط نشان می‌دهد. وقتی $w = 0$ است، دیکدر فوقانی فعال می‌شود و دیگری غیرفعال است. خروجی‌های دیکدر پایینی همگی در 0 خواهند بود و هشت خروجی بالایی مینترم‌های 0000 تا 0111 را تولید می‌کنند. وقتی $w = 1$ باشد، وضعیت فعال شدن معکوس میگردد. خروجی‌های دیکدر پایینی مینترم‌های 1000 تا 1111 را تولید می‌نمایند، در حالی که خروجی‌های فوقانی همه 0 هستند. این مثال حسن ورودی‌های فعال‌ساز را در دیکدرها و دیگر قطعات منطقی ترکیبی نشان می‌دهد. به طورکلی ورودی‌های فعال‌ساز ابزارهای مناسبی برای اتصالات درونی دو یا چند قطعه استاندارد برای گسترش آنها با عملکردی مشابه و ورودی‌ها و خروجی‌های بیشتر است.

پیاده‌سازی مدار منطقی ترکیبی

یک دیکدر، 2^n مینترم را برای n متغیر ورودی تهیه می‌کند. چون هر تابع بولی می‌تواند برحسب جمع مینترم‌ها بیان شود، می‌توان از دیکدر برای تولید مینترم استفاده کرده و با یک گیت OR بیرونی جمع منطقی آنها را تشکیل داد. به این ترتیب هر مدار ترکیبی n ورودی و m خروجی با یک دیکدر n به 2^n و m گیت OR قابل پیاده‌سازی است.

روال پیاده‌سازی یک مدار ترکیبی با دیکدر و گیت‌های OR لازم می‌دارد که تابع بول مدار برحسب



شکل ۲۱-۴ . پیاده‌سازی یک جمع‌کننده کامل با دیکدر

جمع مینترم‌ها بیان شود. سپس یک دیکدر برای تولید همه مینترم‌های حاصل از متغیرهای ورودی انتخاب می‌گردد. ورودی‌های هر گیت OR از خروجی‌های دیکدر برحسب لیست مینترم هر تابع انتخاب می‌گردند. این روال با مثالی که مدار جمع‌کننده کامل را به کار می‌برد تشریح می‌گردد.

با توجه به جدول درستی جمع‌کننده کامل (جدول ۴-۴)، توابع مدار ترکیبی را به صورت مجموع مینترم‌ها بدست می‌آوریم:

$$S(x, y, z) = \Sigma(1, 2, 4, 7)$$
$$C(x, y, z) = \Sigma(3, 5, 6, 7)$$

چون سه ورودی و جمعاً هشت مینترم وجود دارد، به یک دیکدر 3 به 8 خط احتیاج است. پیاده‌سازی در شکل ۲۱-۴ ملاحظه می‌گردد. دیکدر هشت مینترم را برای x ، y و z تولید می‌کند. گیت OR برای خروجی S، جمع منطقی مینترم‌های 1 ، 2 ، 4 و 7 را تشکیل می‌دهد. گیت OR جمع منطقی مینترم‌های 3 ، 5 ، 6 و 7 را برای تولید خروجی C بکار می‌برد.

یک تابع با لیست طویلی از مینترم‌ها نیاز به یک گیت OR با ورودی‌های متعدد دارد. تابعی که k مینترم دارد می‌تواند به فرم متمم خود، F' ، با 2^n-k مینترم نشان داده شود. اگر تعداد مینترم‌های موجود در تابع بزرگتر از $2^n/2$ باشد، آنگاه می‌توان F' را با تعداد مینترم کمتری بیان کرد. در چنین وضعیتی، استفاده از گیت NOR برای تشکیل جمع مینترم‌های F' مزیت دارد. خروجی گیت NOR این جمع را متمم کرده و تولید خروجی نرمال F را خواهد کرد. اگر از گیت‌های NAND ، مثل شکل ۱۹-۴ استفاده شود، آنگاه گیت‌های خروجی در عوض OR باید از نوع NAND باشند. دلیل این است که یک مدار گیتی دو طبقه NAND تابع جمع مینترم‌ها را پیاده‌سازی می‌کند و معادل با مدار دو طبقه AND−OR است.

۹-۴ انکدرها

یک انکدر مداری است که عمل عکس یک دیکدر را انجام می‌دهد. یک انکدر دارای 2^n (یا کمتر) خط ورودی و n خط خروجی است. خطوط خروجی کد دودویی مربوط به مقدار دودویی ورودی را

تولید می‌نمایند.

مثالی از یک انکدر، انکدر هشت هشتی به دودویی است که جدول درستی آن در جدول ۷-۴ داده شده است. این مدار دارای هشت ورودی (یک ورودی برای هر رقم هشت هشتی) و سه خروجی است که عدد دودویی مربوطه را تولید می‌نماید. فرض بر این است که در هر لحظه‌ای از زمان تنها یک ورودی مقدار 1 را داشته باشد.

انکدر را می‌توان با گیت‌های OR که ورودی‌هایشان مستقیماً از جدول درستی تهیه می‌شود، پیاده‌سازی کرد. خروجی z هنگامی 1 است که رقم هشت هشتی ورودی در 1، 3، 5 یا 7 برابر 1 باشد. خروجی y به ازاء ارقام هشت هشتی ورودی 2، 3، 6 و 7، برابر 1 می‌شود. این شرایط را می‌توان با معادلات بولی خروجی بیان کرد.

$$z = D_1 + D_3 + D_5 + D_7$$
$$y = D_2 + D_3 + D_6 + D_7$$
$$x = D_4 + D_5 + D_6 + D_7$$

این انکدر با سه گیت OR قابل پیاده‌سازی است.

انکدری که در جدول ۷-۴ تعریف شد دارای این محدودیت است که در آن در هر لحظه از زمان فقط یک ورودی فعال می‌باشد. اگر دو ورودی به طور همزمان فعال شوند، خروجی ترکیبات نامفهومی را تولید خواهد کرد. مثلاً اگر D_3 و D_6 همزمان برابر 1 شوند، خروجی انکدر 111 خواهد شد زیرا در این حالت هر سه خروجی برابر 1 است. این خروجی نه 3 و نه 6 را نمایش می‌دهد. برای حل این مشکل، مدارهای انکدر باید اولویتی در ورودی ایجاد کنند تا مطمئن شویم که فقط یک ورودی انکد شده است. اگر اولویت بالاتر را با اندیس‌های بالاتر ایجاد نماییم، و اگر در یک زمان D_6 و D_3 برابر 1 شوند، خروجی 110 می‌شود زیرا، D_6 اولویت بالاتری نسبت به D_3 دارد.

مشکل دیگری که در انکدر هشت هشتی به دودویی وجود دارد این است که در آن یک خروجی تمام 0 به ازاء حالتی که همه ورودی‌ها 0 هستند تولید می‌شود. این خروجی برابر با حالتی است که در آن $D_0 = 1$ است. ایراد را می‌توان با تهیه یک خروجی بیشتر حل کرد تا به این ترتیب نشان دهد که حداقل یک ورودی 1 است.

جدول ۷-۴. جدول درستی یک انکدر هشت هشتی به دودویی

ورودی‌ها								خروجی‌ها		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1