

بسمه تعالی



دانشکده کامپیوتر

گزارش کار پنجم آزمایشگاه مدارهای منطقی

## واحد محاسبات و منطق (ALU)

استاد

آقای دکتر شاهین حسابی

دانشجویان

نیکا قادری و مبینا حیدری

دانشگاه صنعتی شریف

پاییز ۱۴۰۲

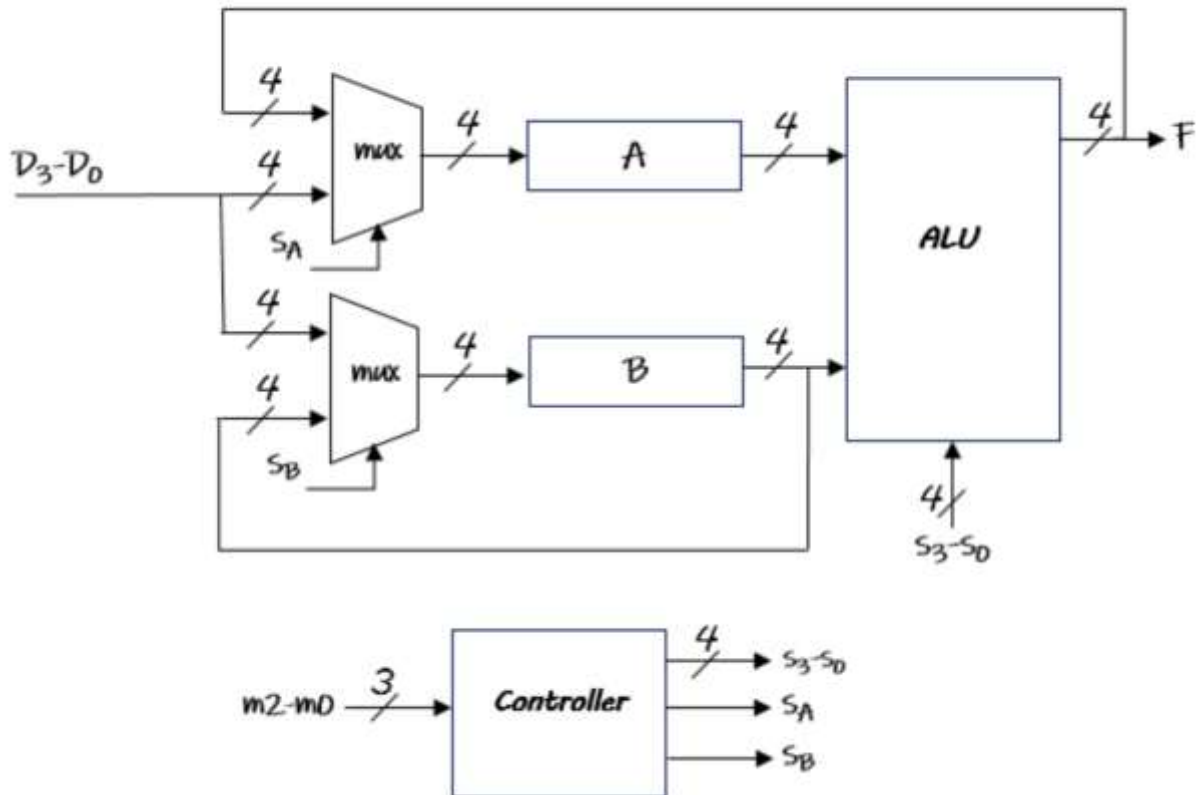
## فهرست مطالب

۳	ازمایش پنجم: واحد محاسبات و منطق (ALU) .....
۳	۵-۱ آشنایی با تراشه ۷۴۱۸۱ .....
۸	۵-۲ ساخت مدار داخلی ALU .....

## ازمایش پنجم: واحد محاسبات و منطق (ALU)

۵-۱ آشنایی با تراشه ۷۴۱۸۱

مداری طراحی کنید که طبق شکل زیر دارای دو ثابت داده A و B ، یک ALU و یک کنترل کننده باشد، به طوری که با دادن کدهای مختلف به ALU ، اعمال مختلف بر روی ورودی ها انجام شود.



مدار باید مطابق جدول زیر با توجه به ورودی های m0-m2 عملیات های خاصی انجام دهد.

جدول ۲- عملیات صورت گرفته در مدار بر حسب ورودی های M0-M2

M2	M1	M0	Operation
0	0	0	$A \leftarrow D_3-D_0$
0	0	1	$B \leftarrow D_3-D_0$
0	1	0	$A \leftarrow A$
0	1	1	$A \leftarrow B$
1	0	0	clear (A)
1	0	1	$A \leftarrow \text{not}(A)$
1	1	0	$A \leftarrow \text{and}(A,B)$
1	1	1	$A \leftarrow \text{add}(A,B)$

وسایل مورد نیاز:

قطعه ۷۴۱۸۱-alu

قطعه ۷۴۱۷۵-multiplexer

قطعه ۷۴۱۷۵-register

Logic state

Logic probe

Push button

گیت ها و ترانزیستور های لازم

قطعه ۷۴۱۷۵:



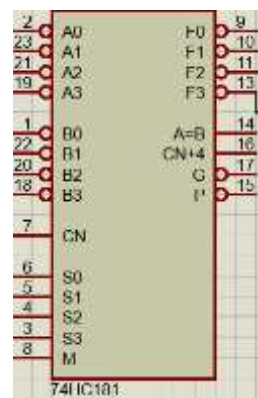
در حقیقت یک d flip flop ۴ بیتی است که قابلیت ذخیره ی داده را دارد. ورودی های آن d0 تا d3 که عددی هستند که می خواهیم لود کنیم. Mr در حکم ریست است و اگر ۰ باشد این قطعه ریست میشود و اگر ۱ باشد میتوان از قطعه استفاده کرد.

قطعه ۷۴۱۵۷:



مثل یک مالتی پلکسر ۲ به ۱ عمل میکند با این تفاوت که ورودی های مالتی پلکسر ۴ بیتی هستند. بسته به ورودی  $A'/B$  یکی از دو عدد ۴ بیتی A یا B در مالتی پلکسر لود میشوند. همچنین E همان enable به فرمت اکتیو لو است که در صورت ۰ بودن میشود از این مالتی پلکسر استفاده کرد.

قطعه 74181 :



یک ALU است که بر حسب ورودی های  $S_0$  تا  $S_3$  و  $M$  و  $C_N$  عملیات های متفاوتی را روی دو ورودی ۴ بیتی  $A$  و  $B$  اش انجام می دهد . که این عملیات ها در جدول بالا مشخص شده بود تنها خروجی های مورد استفاده ی ما از این قطعه  $C_N+4$  یا همان carry out و  $F_0$  تا  $F_3$  یا همان خروجی های عملیات انجام شده هستند.

موارد خواسته شده در جدول زیر نشان داده شده است:

Mode Select Inputs				Active LOW Operands & $F_n$ Outputs		Active HIGH Operands & $F_n$ Outputs	
				Logic	Arithmetic (Note 2)	Logic	Arithmetic (Note 2)
$S_3$	$S_2$	$S_1$	$S_0$	( $M = H$ )	( $M = L$ ) ( $C_n = L$ )	( $M = H$ )	( $M = L$ ) ( $C_n = H$ )
L	L	L	L	$\bar{A}$	A minus 1	$\bar{A}$	A
L	L	L	H	$\bar{A}\bar{B}$	AB minus 1	$\bar{A} + \bar{B}$	A + B
L	L	H	L	$\bar{A} + \bar{B}$	$\bar{A}\bar{B}$ minus 1	$\bar{A}B$	A + $\bar{B}$
L	L	H	H	Logic 1	minus 1	Logic 0	minus 1
L	H	L	L	$\bar{A} + \bar{B}$	A plus ( $A + \bar{B}$ )	$\bar{A}\bar{B}$	A plus $\bar{A}\bar{B}$
L	H	L	H	$\bar{B}$	AB plus ( $A + \bar{B}$ )	$\bar{B}$	(A + B) plus $\bar{A}\bar{B}$
L	H	H	L	$\bar{A} \oplus \bar{B}$	A minus B minus 1	$A \oplus B$	A minus B minus 1
L	H	H	H	$A + \bar{B}$	$A + \bar{B}$	$\bar{A}\bar{B}$	AB minus 1
H	L	L	L	$\bar{A}B$	A plus ( $A + B$ )	$\bar{A} + B$	A plus AB
H	L	L	H	$A \oplus B$	A plus B	$\bar{A} \oplus \bar{B}$	A plus B
H	L	H	L	B	$\bar{A}\bar{B}$ plus ( $A + B$ )	B	( $A + \bar{B}$ ) plus AB
H	L	H	H	$A + B$	$A + B$	AB	AB minus 1
H	H	L	L	Logic 0	A plus A (Note 1)	Logic 1	A plus A (Note 1)
H	H	L	H	$\bar{A}\bar{B}$	AB plus A	$A + \bar{B}$	(A + B) plus A
H	H	H	L	$\bar{A}B$	$\bar{A}\bar{B}$ minus A	$A + B$	( $A + \bar{B}$ ) plus A
H	H	H	H	A	A	A	A minus 1

حال با توجه به دو جدول داده شده جدول حالت را رسم میکنیم

M2	M1	M0	SA	SB	S3	S2	S1	S0	Mode
0	0	0	1	1	d	d	d	d	1
0	0	1	0	0	1	1	1	1	1
0	1	0	0	1	1	1	1	1	1
0	1	1	0	1	1	0	1	0	1
1	0	0	0	1	1	1	0	0	1
1	0	1	0	1	0	0	0	0	1
1	1	0	0	1	1	1	1	0	1
1	1	1	0	1	1	0	0	1	0

حال با توجه به جدول ساده میکنیم:

$$Mode=(m_2m_1m_0)'$$

$$S_a=m_2'm_1'm_0'$$

$$S_b=(m_2'm_1'm_0)'$$

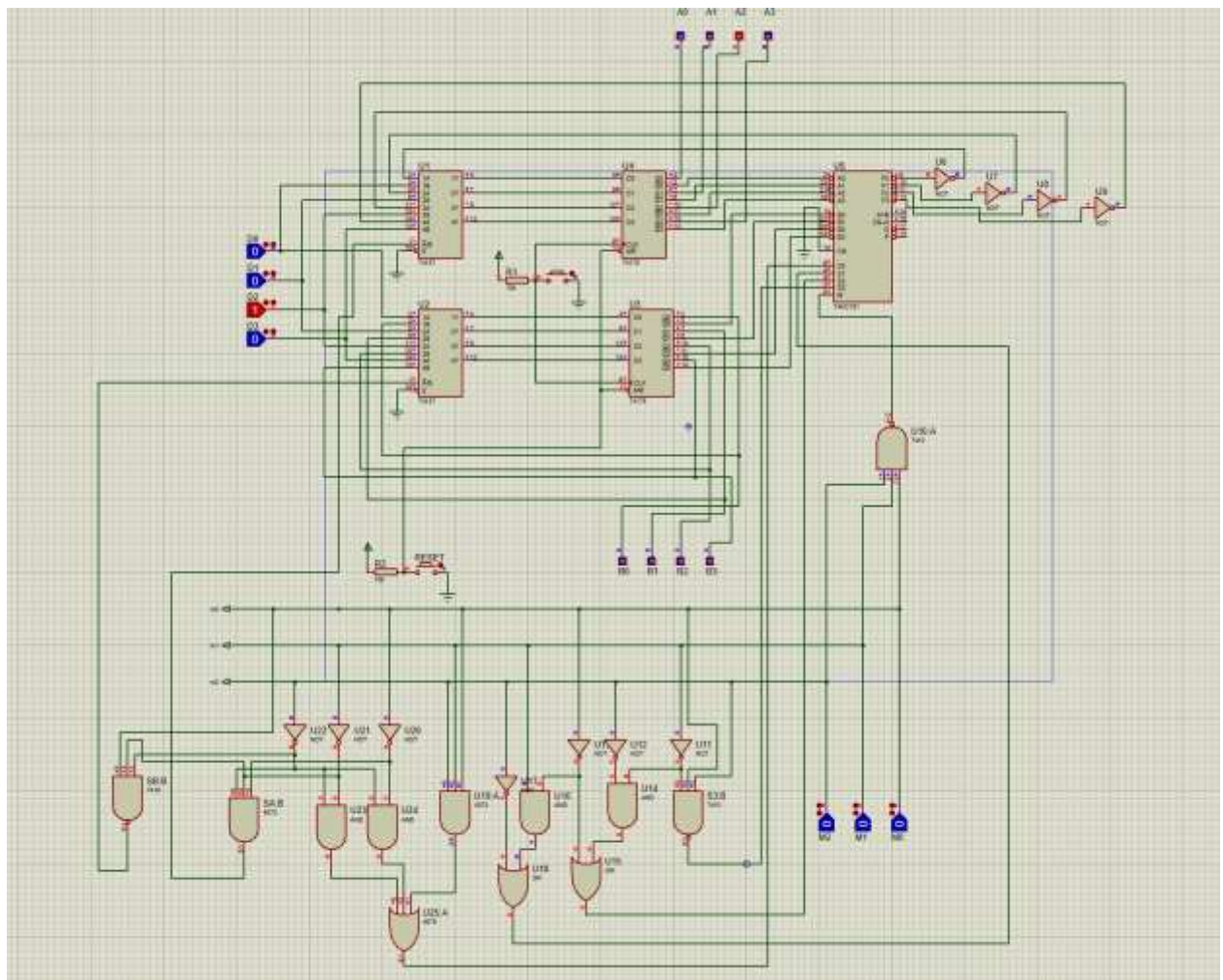
$$S_3=(m_2m_1'm_0)'$$

$$S_2=m_2'm_1'+m_0'$$

$$S_1=m_2'+m_1m_0'$$

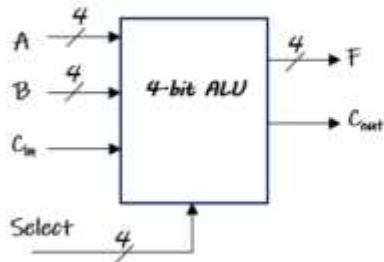
$$S_0=m_2'm_1+m_0'm_2'+m_2m_1m_0$$

در نهایت مدار به صورت زیر است:



## ۵-۲ ساخت مدار داخلی ALU

یک واحد محاسبات و منطق چهاربیتی (ALU) مطابق شکل زیر بسازید.



Operation select					Operation	Function
$S_3$	$S_2$	$S_1$	$S_0$	$C_{in}$		
0	0	0	0	0	$F = A$	Transfer $A$
0	0	0	0	1	$F = A + 1$	Increment $A$
0	0	0	1	0	$F = A + B$	Addition
0	0	0	1	1	$F = A + B + 1$	Add with carry
0	0	1	0	0	$F = A + \overline{B}$	Subtract with borrow
0	0	1	0	1	$F = A + \overline{B} + 1$	Subtraction
0	0	1	1	0	$F = A - 1$	Decrement $A$
0	0	1	1	1	$F = A$	Transfer $A$
0	1	0	0	x	$F = A \wedge B$	AND
0	1	0	1	x	$F = A \vee B$	OR
0	1	1	0	x	$F = A \oplus B$	XOR
0	1	1	1	x	$F = \overline{A}$	Complement $A$
1	0	x	x	x	$F = \text{shr } A$	Shift right $A$ into $F$
1	1	x	x	x	$F = \text{shl } A$	Shift left $A$ into $F$

وسایل مورد نیاز:

قطعه ۷۴۱۵۷-multiplexer

قطعه ۴۰۶۷-multiplexer

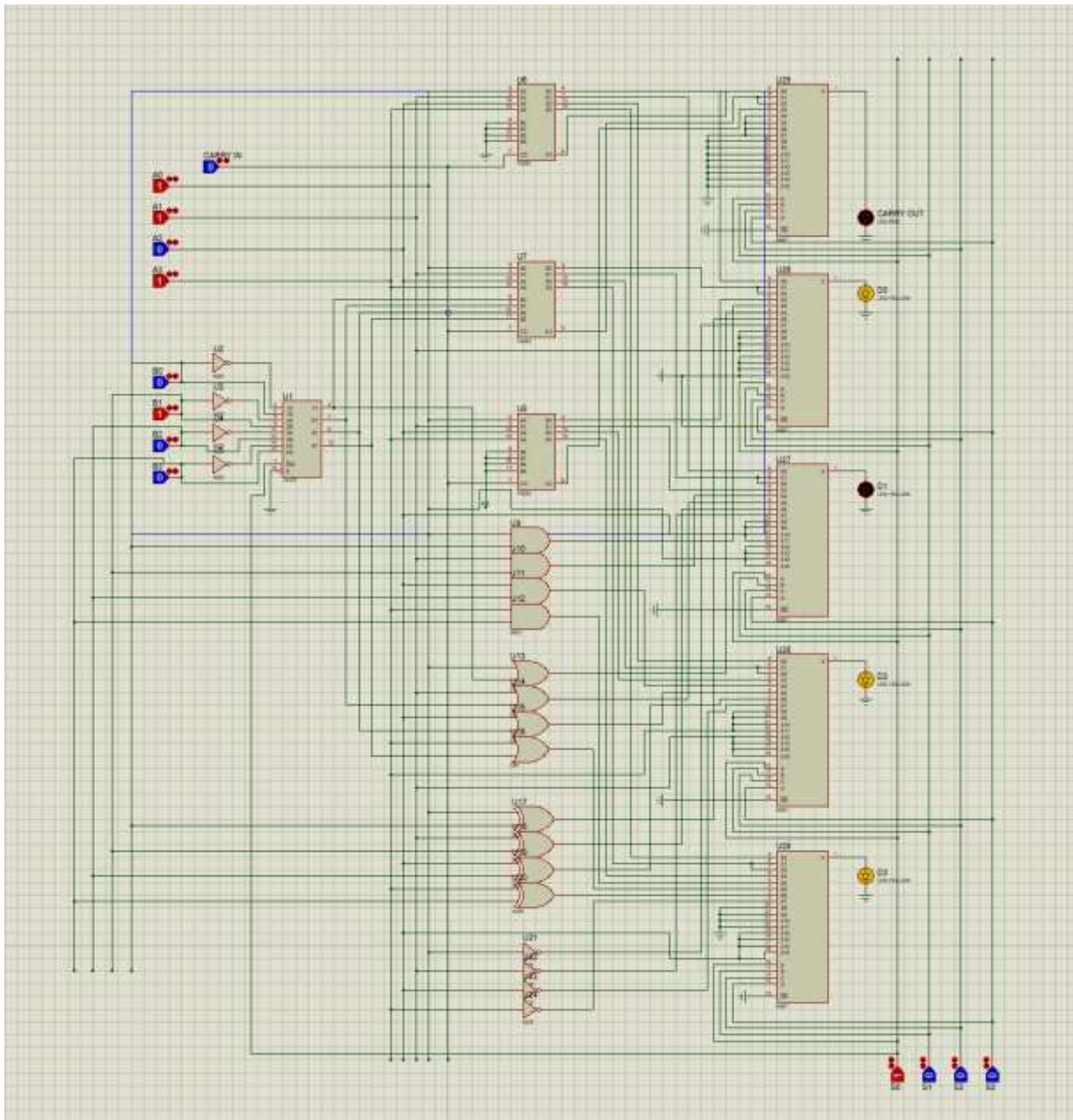
قطعه 74LS283-adder

Led

گیت ها و ترانزیستورهای لازم

شکل مدار در حالت کلی در پایین آمده است

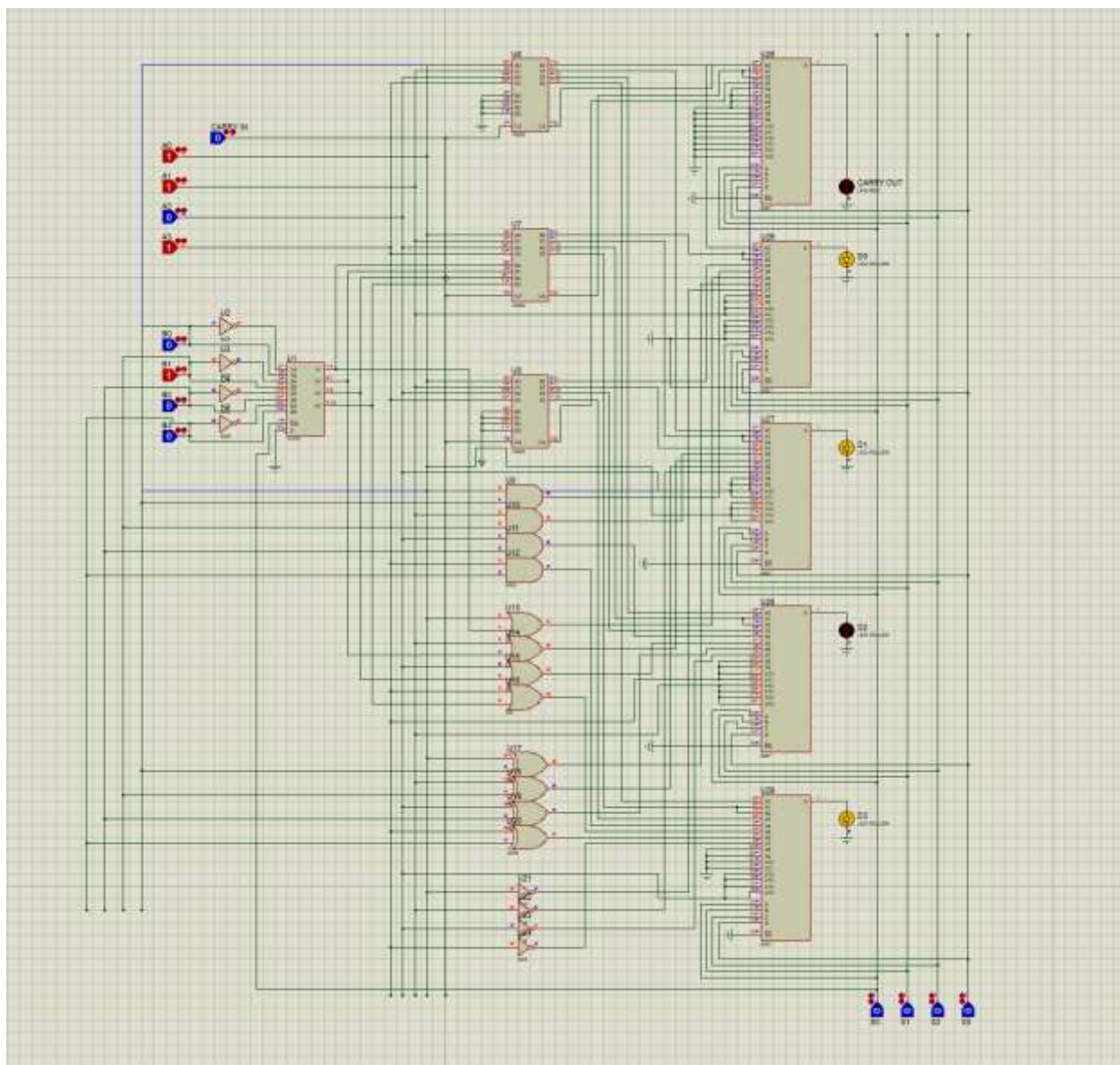




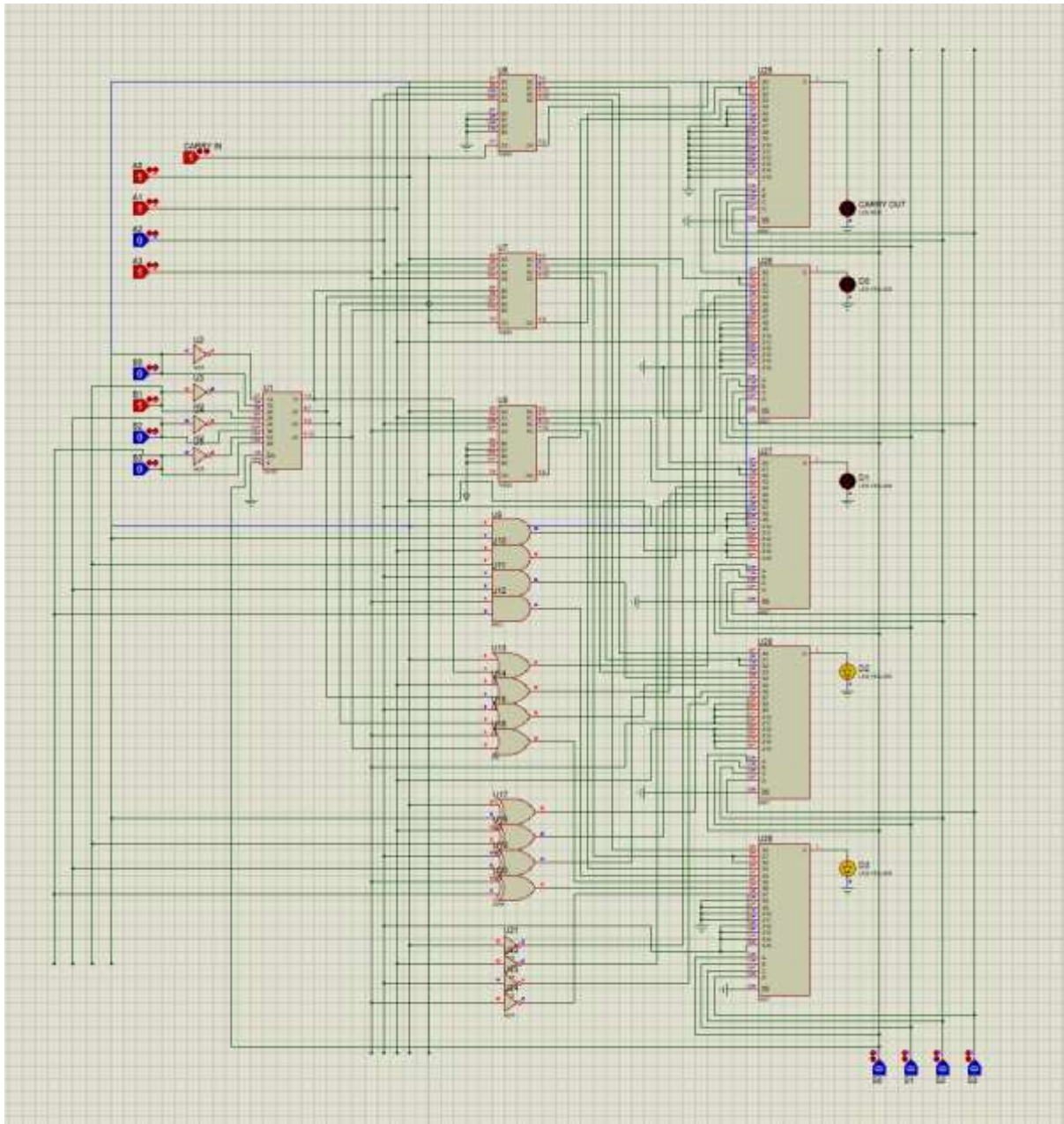
در اینجا D0 D1 D2 D3 خروجی های ما هستند که متصل به چهار multiplexer هستند که ورودی a b c d هر مالتی پلکسر نشان میدهد کدام خط را خروجی میدهد با توجه به جدولی که در پیش تر آمده بود میتوان این کار را کرد به عنوان مثال اگر s0 تا s3 همه صفر باشند و کری صفر باشد a0 a1 a2 a3 در خروجی نشان داده میشود و اگر کری یک باشد A+1 در خروجی می آید که این کار توسط u8 انجام میشود که یک adder می باشد که ورودی A ان مستقیم به A متصل است و carry ان اگر یک باشد جمع ان با یک در خروجی ظاهر میشود و خروجی های این adder را به خط اول چهار مولتی پلکسر متصل کردیم و بقیه نیز به همین صورت انجام میشود

حال چند نمونه را در پایین میبینیم

خود A در خروجی می آید:

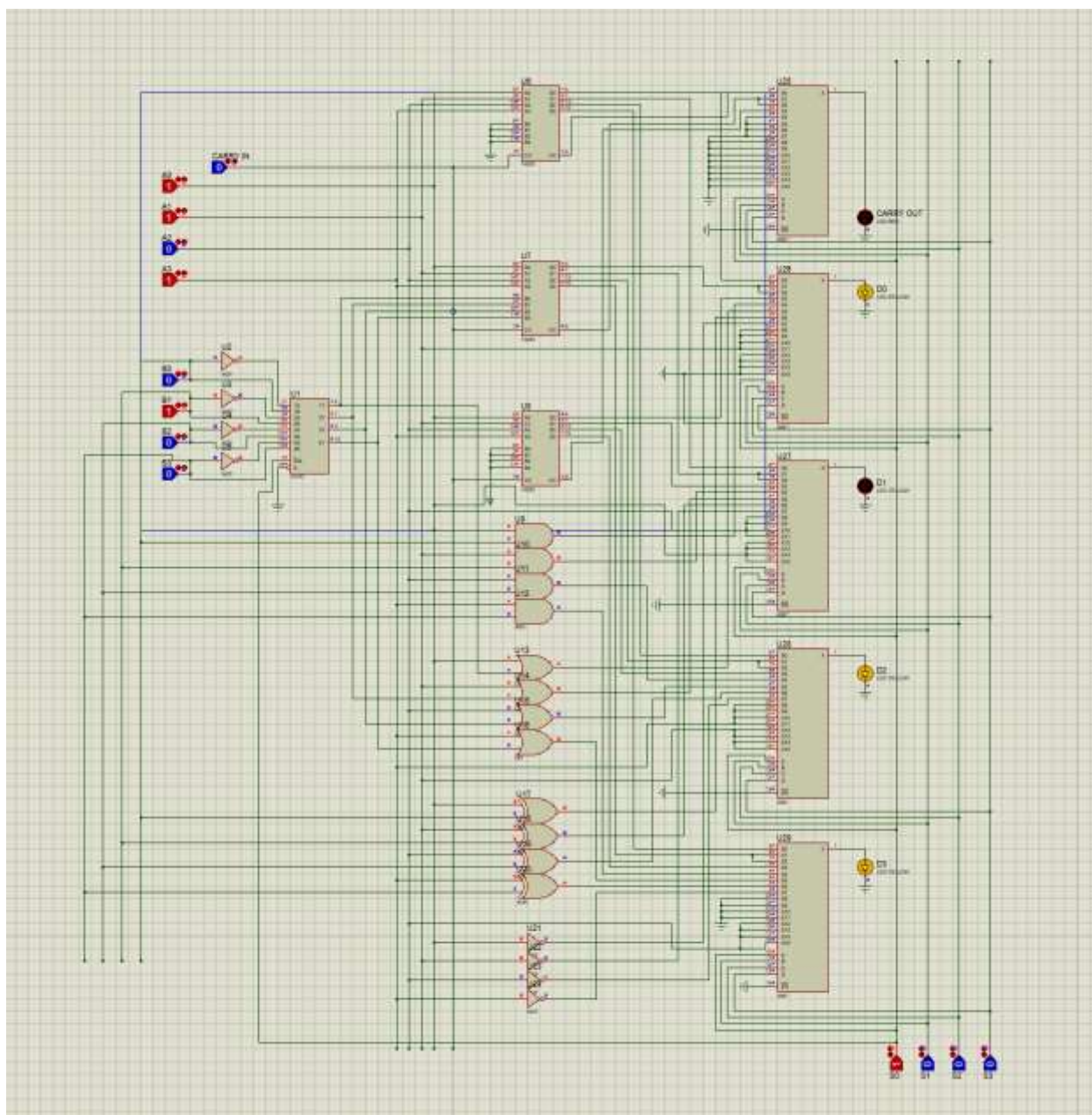


A+1 در خروجی می آید:

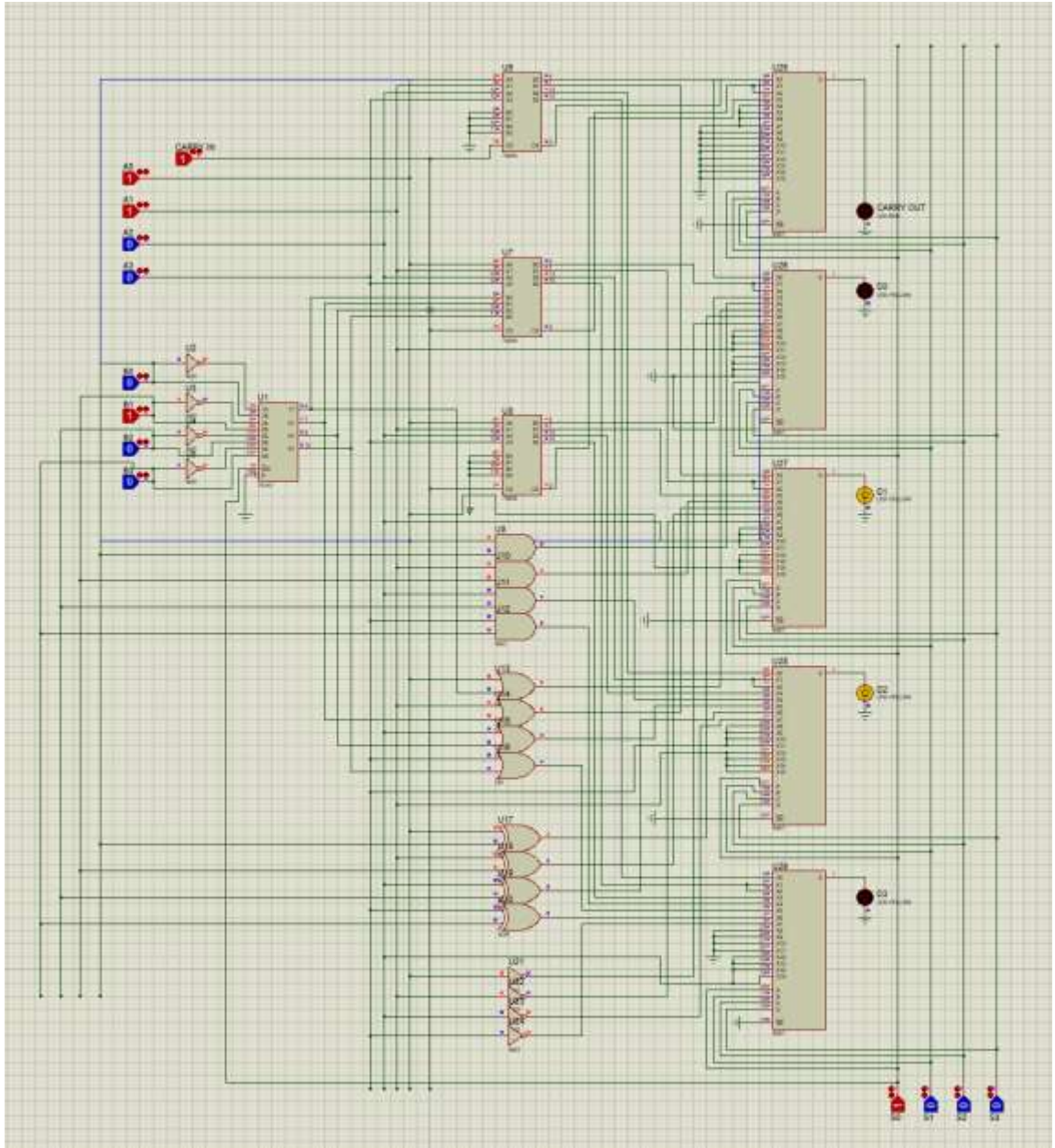


:A+B

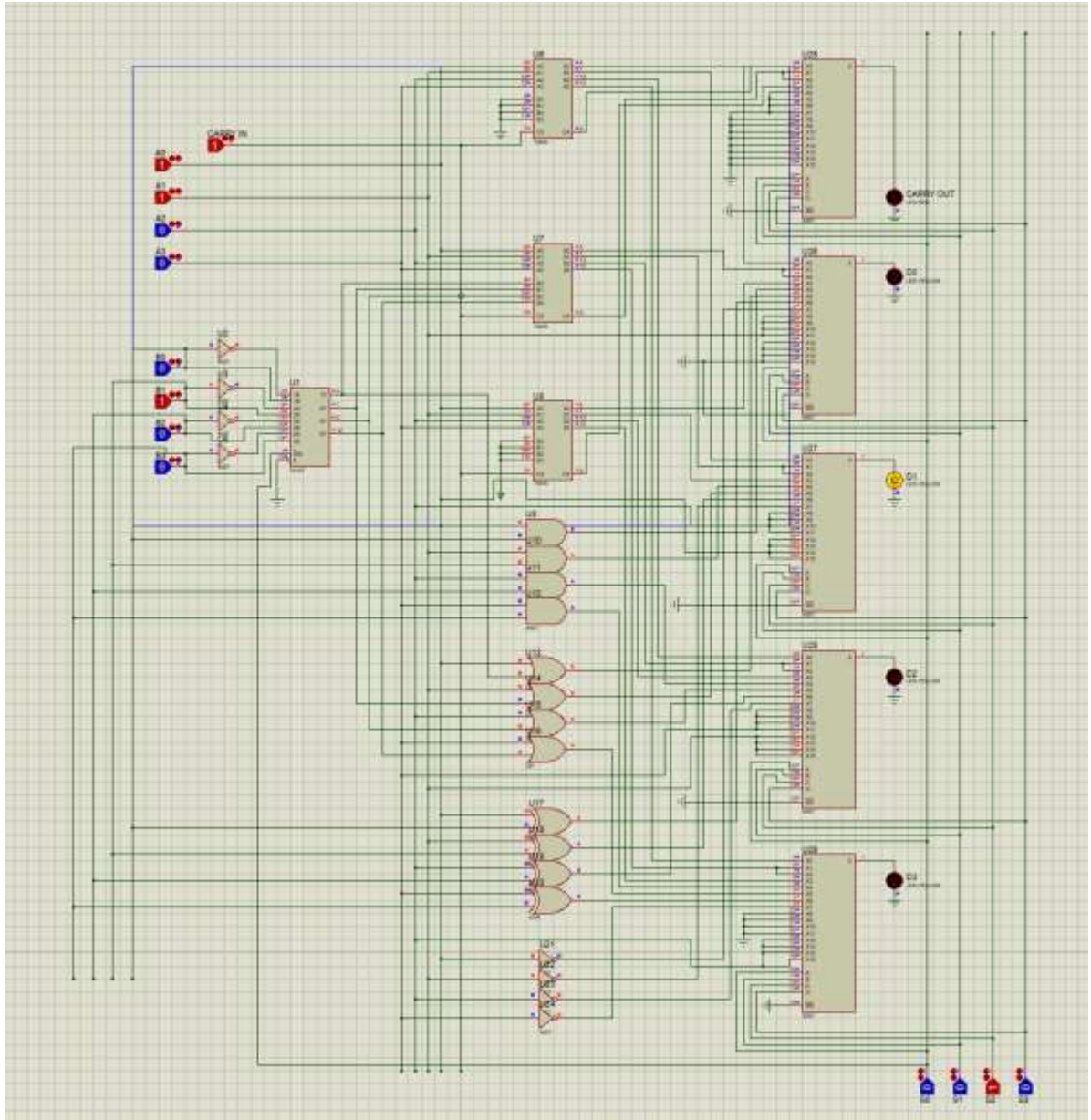




:A+B+1

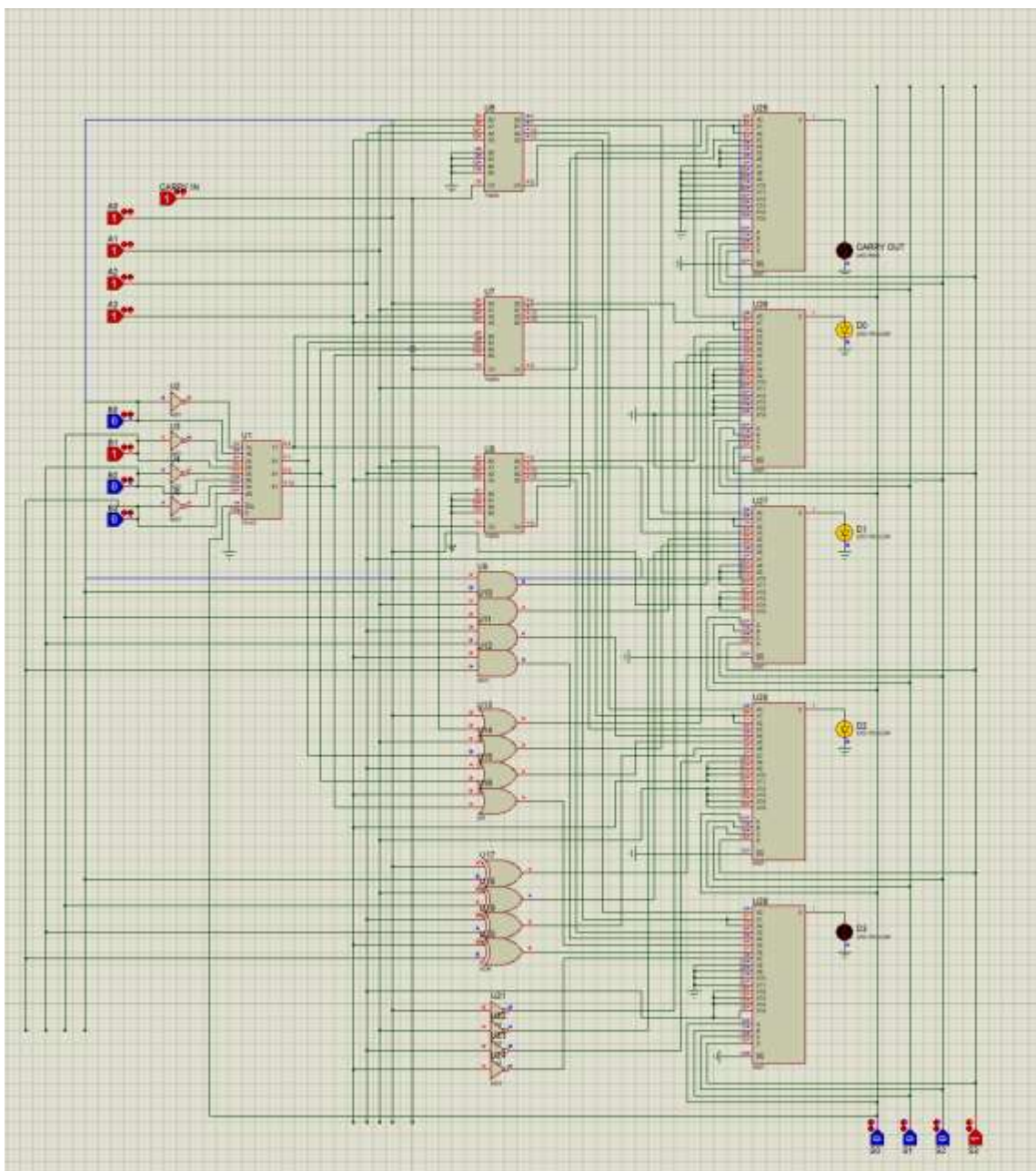


:AandB



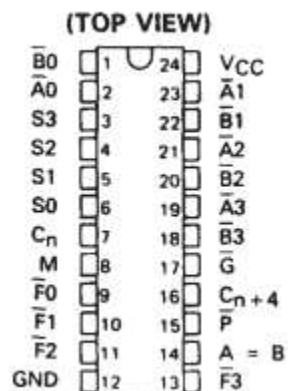
:Shr A





### بخش عملی:

هدف از این بخش آن است که با یک آی سی alu کار کنیم و به طور مثال محاسباتی که در بخش اول این آزمایش را انجام دادیم به طور عملی پیاده سازی کنیم. شکل کلی این آیسی به صورت زیر می باشد:



همان طور که مشاهده می شود، آرسی ۱۳ ورودی می گیرد: ۴ بیت عدد اول (A)، ۴ بیت عدد دوم (B)، ۴ بیت دستور (S) و M که همان مود می باشد. Cn را هم به صفر وصل می کنیم و پس از اتصال vcc و gnd، شروع به ورودی دادن می کنیم. خروجی در بیت های f0, f1, f2 و f3 ذخیره می شود. علاوه بر این ها، یک سری اطلاعات اضافی در مورد عملیات حسابی با استفاده از خروجی های A = B، p و ... قابل دریافت هست که در این آزمایش به آن ها نمی پردازیم.

با استفاده از جدول زیر، در می یابیم که مثلاً برای جمع کردن دو عدد، کافی است که مود و کری را صفر در نظر گرفته و فرمان را به صورت 1001 در ورودی های S وارد کنیم.

ACTIVE-LOW DATA				
SELECTION				M = L: ARITHMETIC OPERATIONS
S3	S2	S1	S0	M = H LOGIC FUNCTIONS
				Cn = L (no carry)
L	L	L	L	F = A
L	L	L	H	F = AB
L	L	H	L	F = A + B
L	L	H	H	F = 1
L	H	L	L	F = A + B
L	H	L	H	F = B
L	H	H	L	F = A ⊕ B
L	H	H	H	F = A + B
H	L	L	L	F = AB
H	L	L	H	F = A ⊕ B
H	L	H	L	F = B
H	L	H	H	F = A + B
H	H	L	L	F = 0
H	H	L	H	F = AB
H	H	H	L	F = AB
H	H	H	H	F = A

†Each bit is shifted to the next more significant position.

عکس های این آزمایش نیز در زیر آورده شده اند:



