



Laborgyakorlat

Logikai áramkörök számítógéppel segített tervezése (CAD)

Bevezetés

A laborgyakorlatok alapvető célja a tárgy későbbi laborgyakorlataihoz szükséges ismeretek átadása, az azokban szereplő korszerű tervezési és vizsgálati eszközök, módszerek első bemutatása.

Napjainkban egyre szélesebb körben alkalmazzák a digitális elektronikus eszközöket. Az eszközök bonyolultsága is egyre nő, nagyon gyakori az olyan alkalmazás, amelynél egyetlen IC tokba több százezer vagy akár több millió kaput integrálnak. Ezek az áramkörök hagyományos papír-ceruza módszerekkel már nem tervezhetők, számítógéppel segített tervezésre (CAD) van szükség. A laborgyakorlatok során a **Xilinx ISE WebPack** fejlesztő rendszert és annak a kapcsolási rajz alapú logikai áramkörtervezését használjuk.

A laboratóriumi gyakorlatok célja, hogy a hallgatók

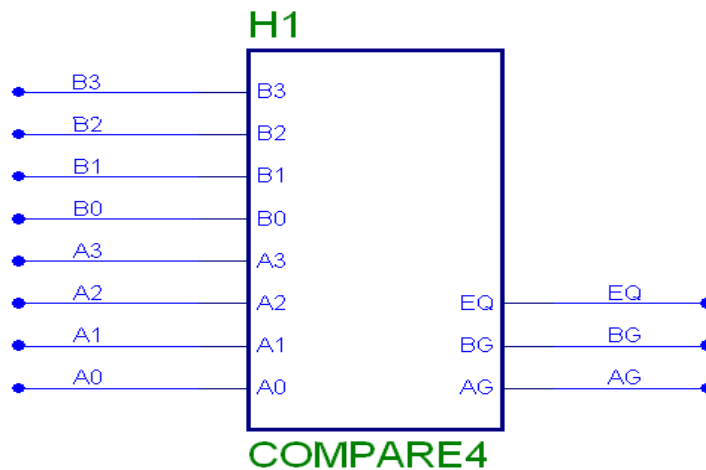
- ismerjék meg a laborgyakorlatok során használt hardver és szoftver eszközöket;
- ismerjék meg a Xilinx ISE Webpack FPGA fejlesztő környezetet és használatát;
- sajátítsák el a kapcsolási rajz alapú logikai áramkörtervezés alapjait;
- önállóan készítsenek el egy kapcsolási rajz alapú példaalkalmazásokat, majd szimulációval és valós hardveren ellenőrizzék azok működését.



(Magnitude) Comparators

A digitális komparátorok 2 azonos bitszélességű bináris értéket hasonlítanak össze, egyszerűbb esetben csak az egyenlőséget jelezve, de az ún. magnitude komparátorok ezen kívül a kisebb, ill. nagyobb relációt is jelzik.

Egy 2-szer 4-bites komparátor blokkrajza:



a két bemenet A(0:3) és B(0:3), az EQ kimenet igaz volta az egyenlőséget (equal), a BGA a B(0:3) a nagyobb voltát, (B greater), az AGB az A(0:3) nagyobb relációt jelenti.

A relációt a legnagyobb helyi érték dönti el, ha egyenlők, akkor a következő helyi érték, stb.

Az áramkör igazságtáblája:

A3	B3	A2	B2	A1	B1	A0	B0	AGB	BGA	EQ
1	0	x	x	x	x	x	x	1	0	0
0	1	x	x	x	x	x	x	0	1	0
=	=	1	0	x	x	x	x	1	0	0
=	=	0	1	x	x	x	x	0	1	0
=	=	=	=	1	0	x	x	1	0	0
=	=	=	=	0	1	x	x	0	1	0
=	=	=	=	=	=	1	0	1	0	0
=	=	=	=	=	=	0	1	0	1	0
=	=	=	=	=	=	=	=	0	0	1

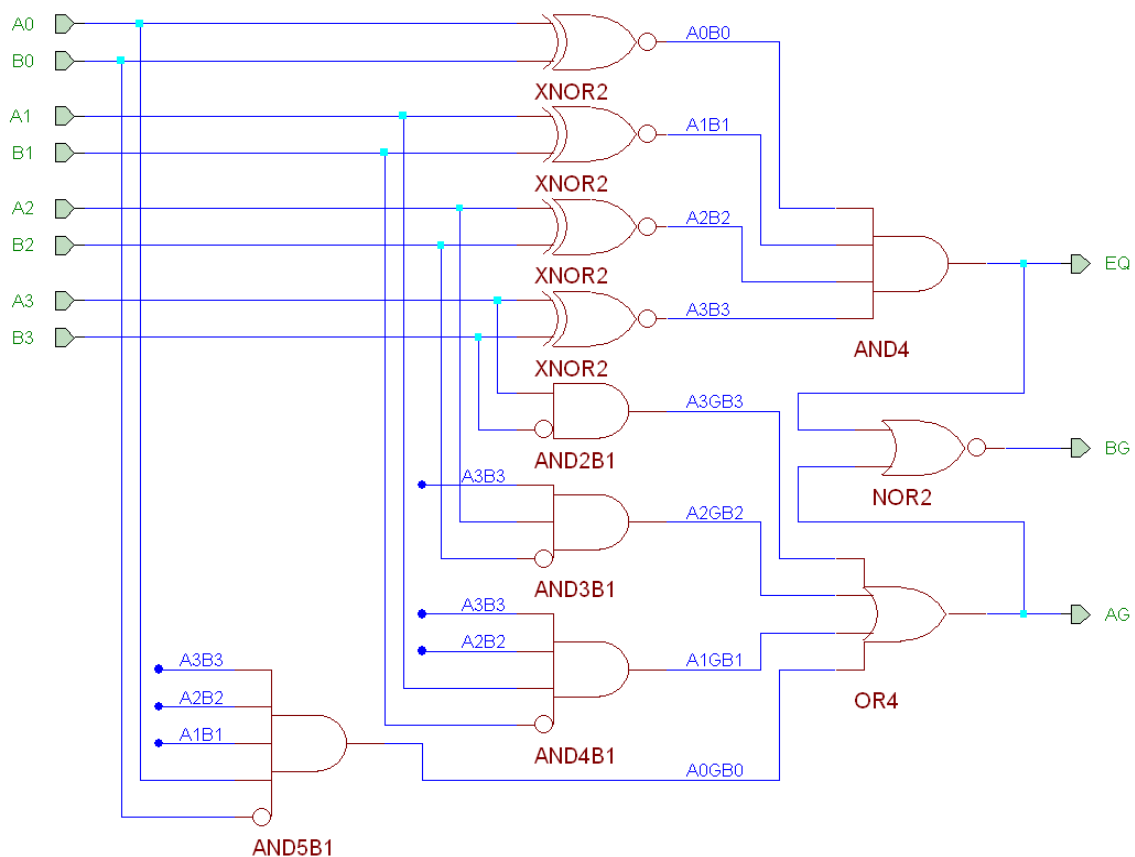


Miután az „A nagyobb”, a „B nagyobb” és „A=B” relációk egymást kizárják, ezért az előző igazságtábla redundáns, egyszerűsíthető:

A3	B3	A2	B2	A1	B1	A0	B0	AGB	EQ
1	0	x	x	x	x	x	x	1	0
=	=	1	0	x	x	x	x	1	0
=	=	=	=	1	0	x	x	1	0
=	=	=	=	=	=	1	0	1	0
=	=	=	=	=	=	=	=	0	1

így, ha $AGB=0$ és $EQ=0$, akkor $BGA=1$.

Egy áramköri megvalósítása:





Encoders

Az encoder (átkódoló, kódoló stb.) egy olyan áramkör, amely az információt valamely formából valamilyen másikba alakítja át. (Természetesen, van sokkal precízebb definíció is).

Mi két átkódolóval foglalkozunk, az egyszerű átkódolóval és a priority átkódolóval.

Az egyszerű átkódoló a sorszámozott egyedi bemeneteket (one-hot) átkódolja a sorszám (v. index) bináris kódjává.

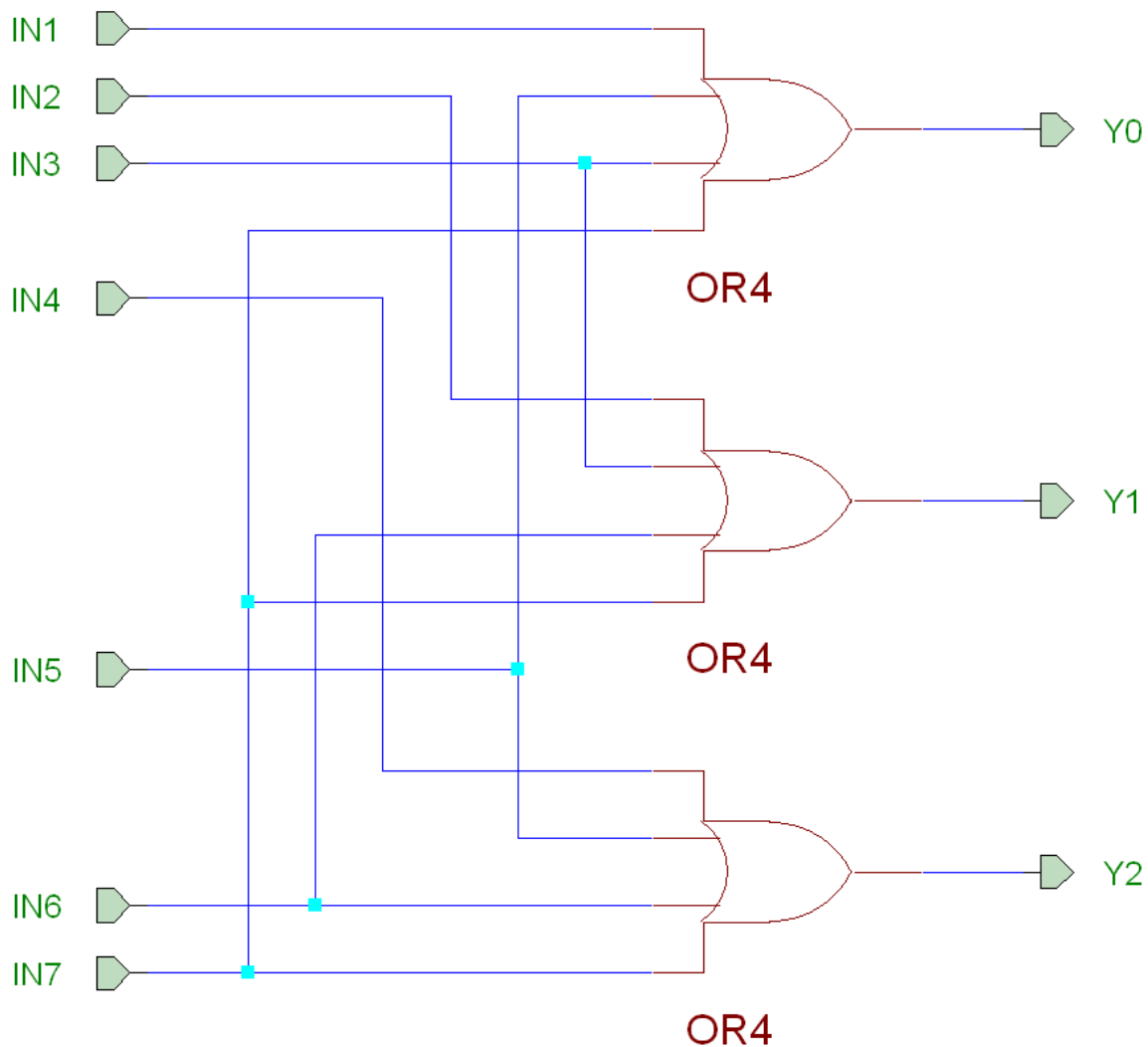
8_to_3 encoder

Az igazságtáblázata:

/	Y2	Y1	Y0
In0	0	0	0
In1	0	0	1
In2	0	1	0
In3	0	1	1
In4	1	0	0
In5	1	0	1
In6	1	1	0
In7	1	1	1



Egy áramkörü megvalósítása:



A bemenetek közül egy és csakis egy lehet aktív. Az áramkör nem tudja megkülönböztetni azt, hogy egy vagy több bemenet aktív egyszerre.



A prioritás kódoló (priority encoder)

A prioritás kódoló esetében egyszerre több bemenet is aktív lehet, a kimeneten a legnagyobb prioritású aktív bemeneti jel indexe jelenik meg bináris kódolással. Ezen kívül van egy kimenet, amely jelzi, hogy legalább egy bemenet aktív.

4 bites prioritás kódoló

Igazságtáblája:

I n0	I n1	I n2	I n3	O ut1	O ut0	I T
0	0	0	0	x	x	0
1	0	0	0	0	0	1
x	1	0	0	0	1	1
x	x	1	0	1	0	1
x	x	x	1	1	1	1

Out0

In0In1 \ In2In3	00	01	11	10
00	x	1	1	0
01	1	1	1	0
11	1	1	1	0
10	0	1	1	0

Out1

In0In1 \ In2In3	00	01	11	10
00	x	1	1	1
01	0	1	1	1
11	0	1	1	1
10	0	1	1	1



$$\text{Out0} = \text{In1} * \text{NIn2} + \text{In3}$$

$$\text{Out1} = \text{In2} + \text{In3}$$

$$\text{IT} = \text{In0} + \text{In1} + \text{In2} + \text{In3}$$

8 bites prioritás kódoló

Igazságtáblája:

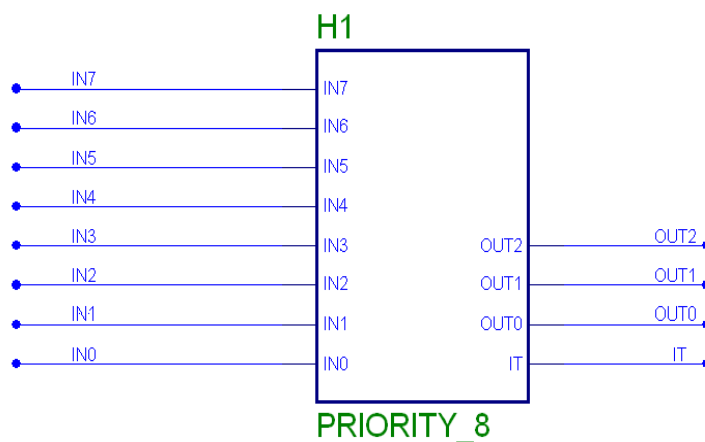
In0	In1	In2	In3	In4	In5	In6	In7	Out2	Out1	Out0	IT
x	x	x	x	x	x	x	1	1	1	1	1
x	x	x	x	x	x	1	0	1	1	0	1
x	x	x	x	x	1	0	0	1	0	1	1
x	x	x	x	1	0	0	0	1	0	0	1
x	x	x	1	0	0	0	0	0	1	1	1
x	x	1	0	0	0	0	0	0	1	0	1
x	1	0	0	0	0	0	0	0	0	1	1
1	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	x	x	x	0

$$\text{Out2} = \text{In4} + \text{In5} + \text{In6} + \text{In7}$$

$$\text{Out1} = \text{In2} * (\text{In3} * \text{NIn4} * \text{NIn5}) + \text{In3} * (\text{NIn4} * \text{NIn5}) + \text{In6} + \text{In7}$$

$$\text{Out0} = \text{In1} * (\text{NIn2} * \text{NIn4} * \text{NIn6}) + \text{In3} * (\text{NIn4} * \text{NIn6}) + \text{In5} * (\text{NIn6}) + \text{In7}$$

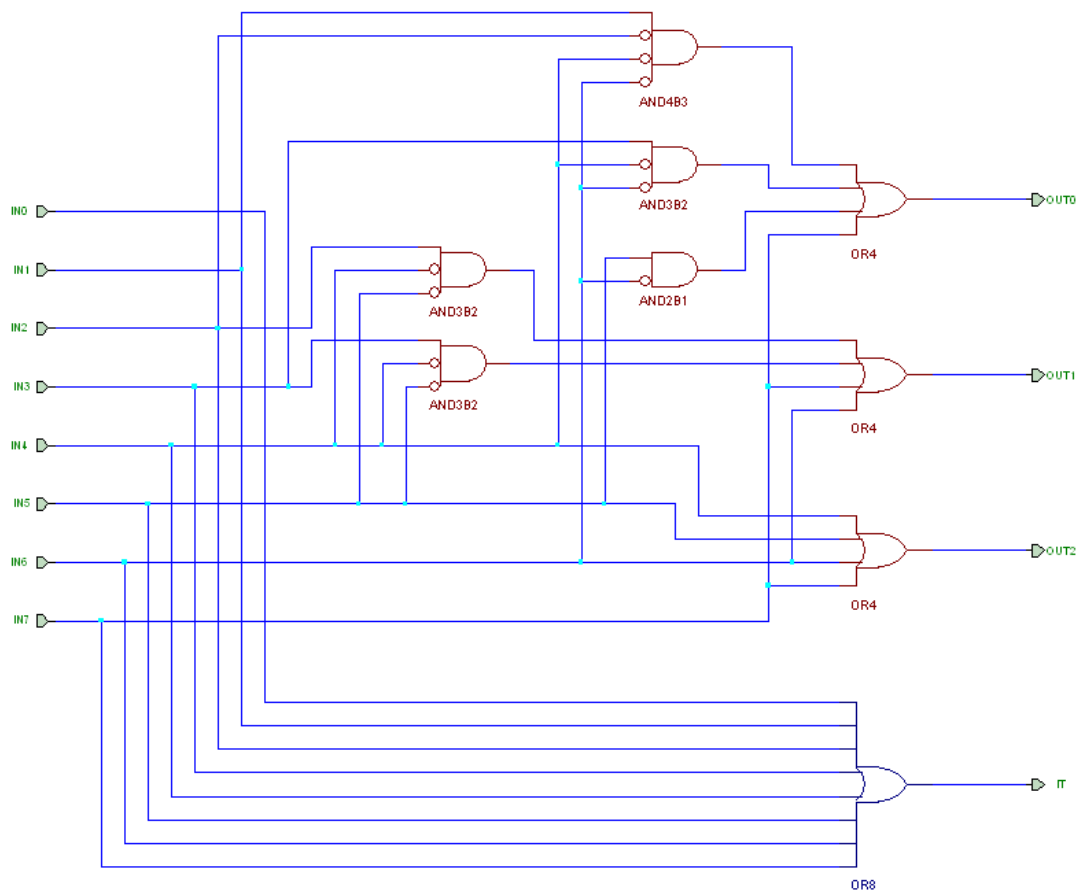
$$\text{IT} = \text{In0} + \text{In1} + \text{In2} + \text{In3} + \text{In4} + \text{In5} + \text{In6} + \text{In7}$$



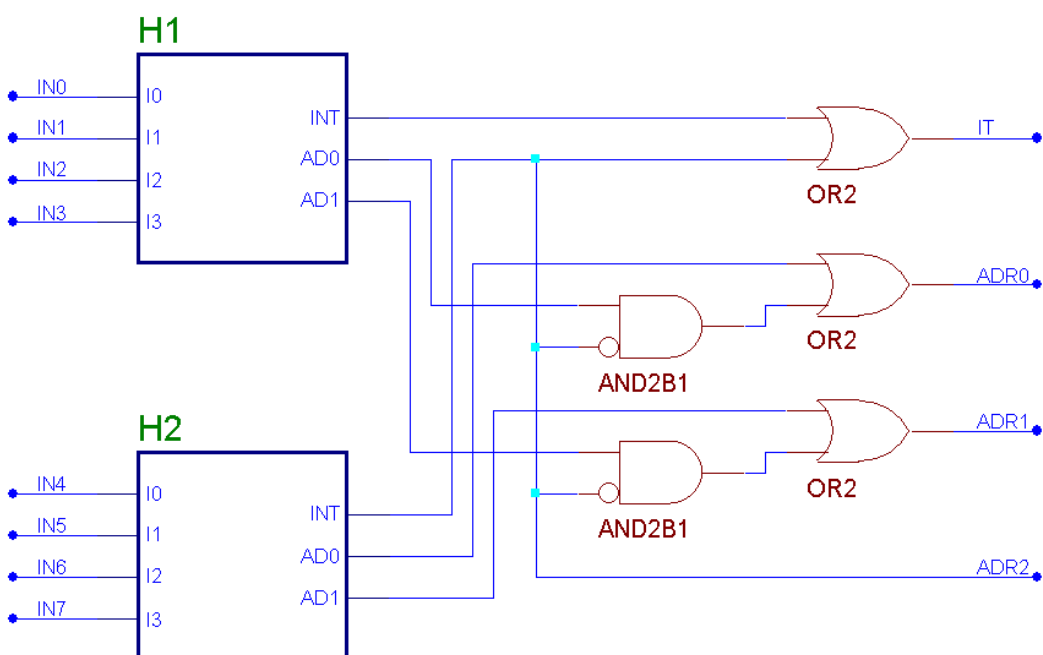
a 8 bites prioritás kódoló blokkrajza.



Egy áramkörti megvalósítása:



A 8 bites prioritás kódoló megvalósítása 2 darab 4 bitesből összekapcsolva (kaszkádosítva).





Decoders, ill. demultiplexer (DMPX)

A dekóder a kódoló inverze, visszaalakítja az információt a kezdetire. Az egyszerű dekóder az **n bites** binárisan kódolt bemeneti adatot átalakítja a 2^n darab kimenetén megjelenő 2^n -ből egy és csak egy aktív kódú (one-hot) kimeneti jelre.

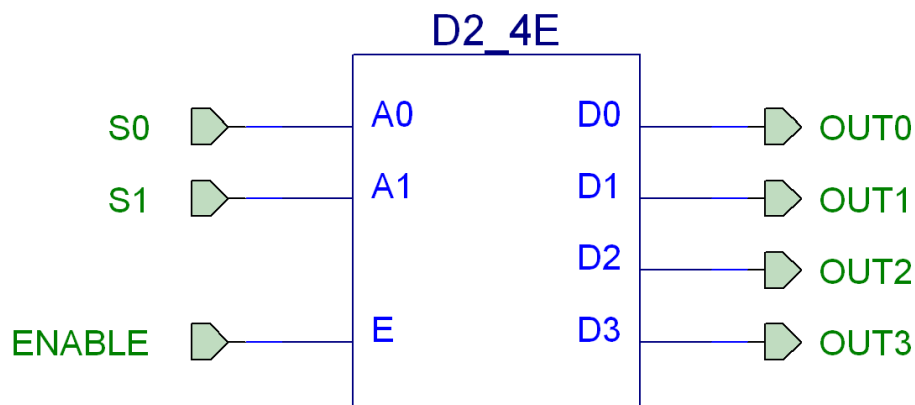
A dekóder ún. demultiplexerként (jel szétosztó) viselkedik, ha az EN logikai jelet adatbemenetként és az S(1:0) jelet címbemenetként értelmezzük. Az ilyen a demultiplexer az adatbemenetén (EN) érkező logikai jelet a megcímzett kimenetre (OUT(3:0)) juttatja el.

4-to-1 dekóder, engedélyező jellel

Igazságtáblája:

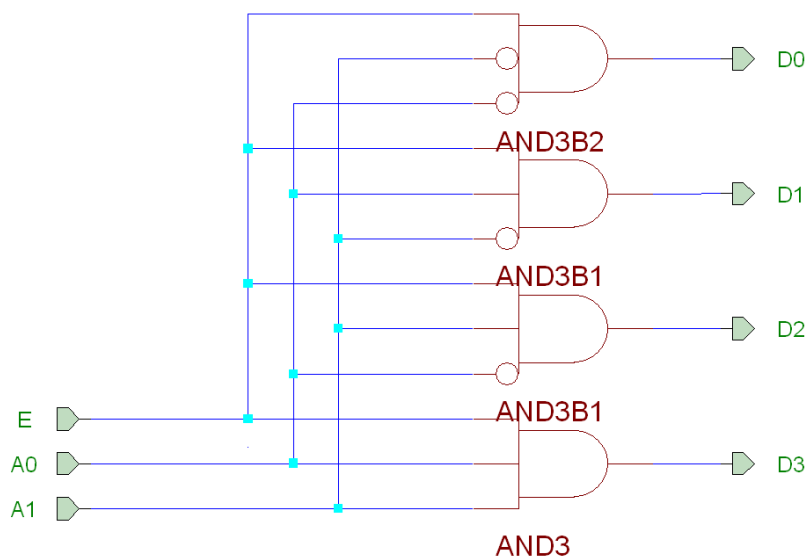
EN	S1	S0	OUT3	OUT2	OUT1	OUT0
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

A blokkrajza (több ilyen egységet össze lehet kapcsolni bővítés céljából, bővíthető címzés szerint és adat szélesség szerint is DMPX esetén):





A 4-to-1 dekóder, engedélyező jellel egy áramköri megvalósítása, amely egyúttal a 4-ről 1-re demultiplexer kapcsolási rajza is:





Laborfeladat:

Tervezzen 2x2 bites komparátorból 2x4 bitest!

A 2x2 bites komparátor

bemenetei: A0, A1, B0, B1 és az EQin, AGin (a két utóbbit a bővítéshez használjuk fel)

kimenetei: EQout, AGout.(a bővítéshez használjuk fel, a láncba kötött bővítést

kaszkádosításnak is nevezzük)

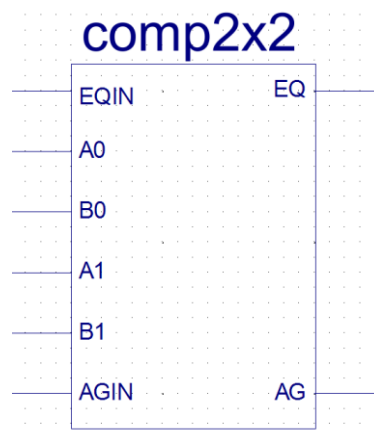
A 2x4 bites komparátor először a magasabb helyi értékű (MSB) 2 db bemeneti bitet hasonlítja össze és ezután az alacsonyabb 2 db-ot (LSB). Lehetséges más sorrendben végezni az összehasonlítást (ekkor más lesz a terv).

Az igazságtábla alapján tervezze meg az áramkört. A legfelsőbb szintű rajzot top.sch-nak nevezze el! A szimulációhoz használja a kiadott tesztfájlt (testbench: tb_.vhd).

Használja fel a BASYS2 demó panel itt felsorolt portjait:

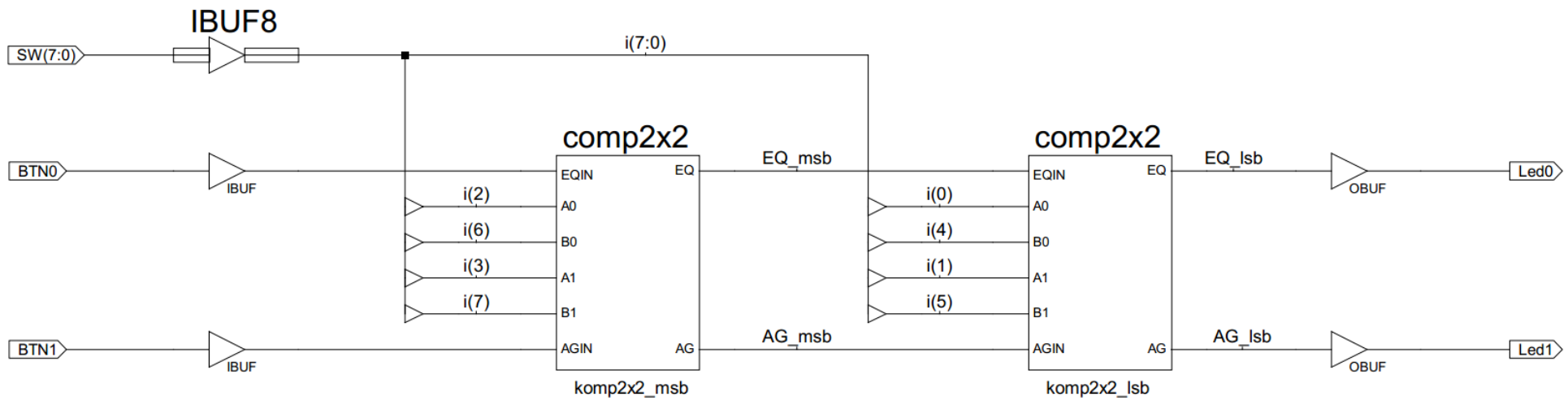
SW7	SW6	SW5	SW4	SW3	SW2	SW1	SW0	BTN1	BTN0	LED1	LED0
B3	B2	B1	B0	A3	A2	A1	A0	AGin	EQin	AGout	EQout

A 2x2 bites komparátor interfész jelei:





Lab_04: 4x4-es komparator - 2x2-es komparator modul felhasználásával



Vígh Tamás ÓBUDAI EGYETEM, 2015



Alkalmazandó műszerek és eszközök

- PC számítógép
- Digilent Basys2 Spartan-3E FPGA mérőpanel
- Digilent Adept konfiguráló szoftver

Hivatkozások, felkészüléshez ajánlott irodalom

- [1] FPGA fejlesztés a Xilinx ISE Webpack-ben, Elektronikus formában a tantárgy honlapján
- [2] Digilent Basys2 Board Reference Manual, Elektronikus formában a tantárgy honlapján
- [3] Spartan-3E Libraries Guide for Schematic Designs, Elektronikus formában a tantárgy honlapján
- [4] Kóré László: Digitális elektronika I. BMF 1121
- [5] Arató Péter: Logikai rendszerek tervezése, Tankönyvkiadó