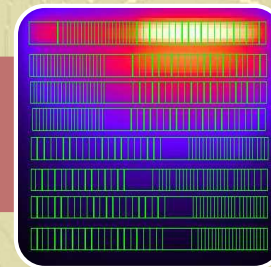
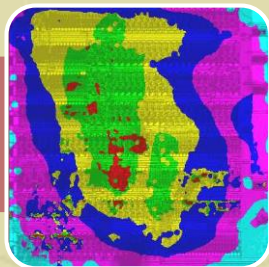
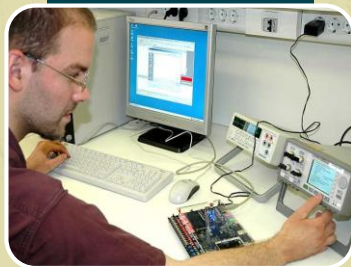
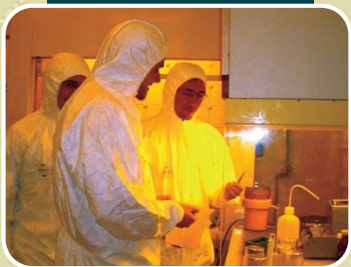


Budapesti Műszaki és
Gazdaságtudományi
Egyetem

Elektronikus Eszközök Tanszéke

Mikroelektronikai tervezés

Tervezőrendszerek



Egy kis történelem

- ▶ Hogyan is terveztek digitális IC-t pl. az 1970-es években?
- ▶ MOS Technology 6502
 - 16 μ m csíkszélesség, kb. 1..2MHz-es órajel, NMOS technológia
 - 3510 tranzisztor, 21mm² chip, 5V tápfeszültség, 16 bites címbusz, 8 bites működés. (összesen 5 regiszter A, X, Y, SP, PC)
 - Ára: \$25
 - (megtudnánk csinálni modern eszközökkel kb. 7000 tranzisztorból?)
- ▶ Kézzel tervezett mikroprocesszor
- ▶ Legendás számítógépekben használták...
 - Commodore 64 (6510, +6 I/O port.), Apple II. stb.



A „sötét középkor”

▶ Digitális tervezés

- Szimulátorok nélkül..., vagy kezdetleges in-house szimulátorokkal
- Kapuszinten

▶ Kapuk tervezése

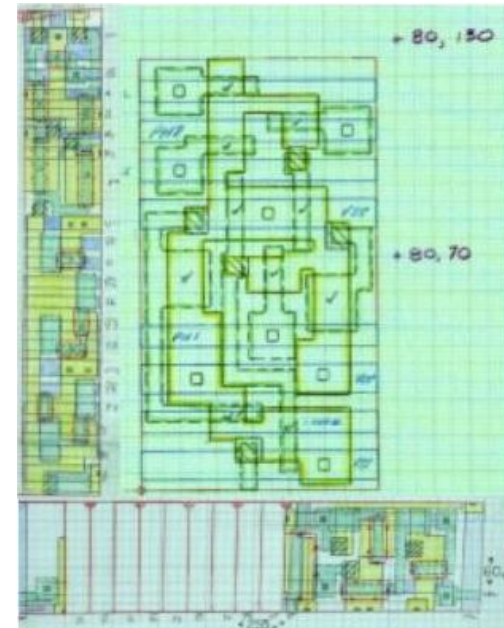
- „kockás papír”
- Áramköri és layout szinten.
- Látható, hogy standard cellás a tervezés...

▶ Maszkok előállítása

- „rubilith” fólia, késsel vágdosva.
- Majd optikai módon kicsinyítve.

▶ DRC/LVS

- Ember által...



Reprezentációs szintek digitális tervezés esetén

Szint	Reprezentáció	Szimulátor
Rendszerszintű tervezés (ESL)	forráskód, a rendszer leírása valamilyen, erre alkalmas nyelv segítségével.	Rendszerszintű szimulátor
RTL szintű leírás	forráskód, a hardvert és a kiválasztott architektúrát egyértelműen meghatározó, szintetizálható HDL	Logikai szimulátor
Strukturális leírás	HDL forráskód, a technológiában rendelkezésre álló kapuk és összeköttetések	Logikai szimulátor
Fizikai tervezés	Layout, valamilyen geometriai leíró nyelven	Áramköri szimulátor



A (mikro)elektronikai cad

- ▶ avagy EDA (electronic design automation)
- ▶ **adatbázisok:** *az IC adott tervezési fázisnak megfelelő reprezentációi (pl. HDL, kapcsolási rajz, layout)*
- ▶ **szimulációs programok:** *egy adott reprezentációban a működés vizsgálatára (pl. tranzistor szintű szimuláció, logikai szimuláció, stb.)*
- ▶ **editorok:** *egy adott reprezentáció **kézi** létrehozására (pl. áramkör editor, layout editor)*
- ▶ automatikus konverterek a reprezentációk között:
 - **szintézis:** magasabb szint → alacsonyabb szint
 - **visszafejtés:** alacsonyabb szint → magasabb szint



A felsorolt elemek nem utalnak semmiféle realizációs módszerre!

Ez miért lehetséges?

- ▶ IC technológiák tervezése ↔ alkalmazás tervezés *élesen szétválasztva.*
- ▶ Kapocs közöttük: *tervezési szabályok, eszközparaméterek...*

Ennek milyen következményei vannak?

- ▶ Nyílt tervezőrendszerek lehetségesek (ugyanaz a szoftver tetszőleges technológiára, realizációs módra).
- ▶ Tetszőleges eszköz használható, bármelyik gyártó tervezőrendszerébe beépíthető egy másik gyártó/vagy saját eszköz, több kevesebb munka befektetésével...
- ▶ Tetszőleges technológiára tervezhetünk, ha beszereztük az ún. PDK-t.
 - Process Design KIT



Az adatbázis reprezentációja

▶ Ipari standard formátumok

- Pl. netlist – spice formátumú netlist
- Cellakönyvtár – TLF (timing library format)
- Layout – GDS2
- LEF – library exchange format

▶ Open Access Formátum (.oa)

- Ipari standard, a nagy gyártók megegyezésén alapul.
- Folyamatos az áttérés



Tervezőrendszerek elemei

▶ Áramkörbevitel

- HDL (Verilog, VHDL stb.)
 - viselkedési leírás (Verilog, VHDL, SystemC, C, bármi 😊)
 - strukturális leírás (Verilog, VHDL)
- Grafikus megadás (strukturális)

▶ Szimuláció (minden absztrakciós szinten)

- rendszer, kapu szintű logikai, áramköri
- megjelenítő eszközök
- koncepcionális tervezés, fizikai tervek ellenőrzése

▶ Szintézis (átjárás az egyes absztrakciós szintek között)

- Magas szintű szintézis
 - Viselkedési leírás → RTL leírás
- Logikai szintézis
 - RTL leírás → strukturális leírás
- Layout szintézis
 - Strukturális leírás → maszkok leírása



Editorok

► Text editor 😊

- Ez minden szinten működik
- Ugyanis mind a szimulátorok, mind a szintézerek alapvetően stdin/stdout programozási modellt követik
- Bár a tervezőrendszerek belső ábrázolása bináris adatbázis, az eszközök általában szöveges leírásból dolgoznak, a konverterek rendelkezésre állnak.
 - A GUI nagyon sok esetben tulajdonképpen szövegfile-okat állít elő...
 - Ez nagy előny akkor, ha kézzel javítjuk ki a hibásan működő GUI-t.
 - Vagy az ismétlődő munka jól scriptelhető!
- Egyes szintekhez léteznek vizualizációs eszközök
 - a tervező termelékenységet segíti
 - Kódkiegészítés, szintaxis kiemelés és ellenőrzés.
 - Kódfedettség
 - Automatikus tesztgenerálás
 - Gyakran használt elemek, pl. állapotgép vizuális megjelenítése és szerkesztése



Editorok

▶ Sémakerkesztő (a továbbiakban schematic editor)

- Kapcsolási rajzok ill. blokkok bevitele, megtekintése, szerkesztése
- Hierarchikus tervezést támogatja
- Szimbólumok, többlapos tervek stb.

▶ Layout editor

- Feladat: alapvetően geometriai alakzatok létrehozása különböző rétegeken
- (ez azért nem túl barátságos)
- parametrikus makrók állnak rendelkezésre pl. egy tranzisztor layout elkészítéséhez.
- A schematic alapján segítséget ad az összeköttetések megvalósításához.
- (kvázi) automatikusan elhelyezi az alkatrészeket. (főleg, ha a kapcsolási rajzon biz. kényszereket megadtunk. – ez leginkább analóg tervezés esetén fontos.
- Egyéb, kiegészítő eszközök, pl. tranzisztorok összevonása, több rétegből álló ún. multipart path pl. guard ring elkészítése stb.

Szintézerek - HLS

▶ Bemenet

- magas szintű leírás
 - C, SystemC, SystemVerilog, VHDL, Matlab kód stb.
- Technológia file
 - Nem valódi kapuk, hanem csak információ, hogy az egyes nyelvi szerkezetek leképezése „mennyibe” kerül

▶ Kényszerek

- Interface
- Architektúra választás
- Időzítés, erőforrás használat (terület), teljesítmény
- (ezen a szinten BECSÜLT értékek)

▶ Kimenet: RTL szintű leírás

- Verilog vagy VHDL választhatóan.



Szintézer – logikai szintézis

▶ Bemenet

- RTL kód
- A cellakönyvtár leírása (hdl nyelven)
- A cellakönyvtár időzítési és teljesítményadatai

▶ Kimenet

- Strukturális HDL, ami már csak a cellakönyvtárbeli elemeket tartalmazza. (valójában ez már netlistnek tekinthető)

▶ Kényszerek:

- Időzítés, terület, teljesítmény
- (alapesetben becsült)
- Együtt tud működni a fizikai elhelyezést végző layout szintézis eszközzel, így jobb minőségű kimenetet szolgáltat.



Szintézer – layout szintézis

▶ Bemenet

- Struktúrált hdl leírás
- A cellakönyvtár layoutja
- Technológia szabályok
 - rendelkezésre álló huzalozási rétegek

▶ Kimenet

- Layout

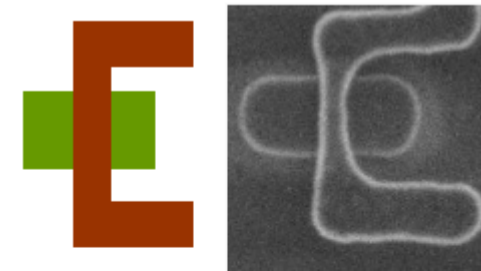
▶ Kényszerek

- Pin elrendezés
- Floorplan
- Oszlopok, sorok, táp
- Órajel elosztó hálózat

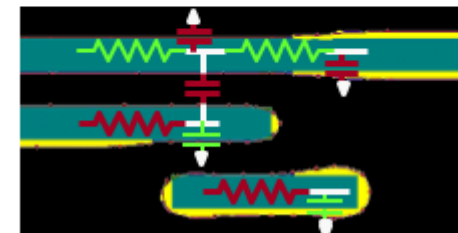


Ellenőrző eszközök

- ▶ DRC - Design rule check
 - Geometriai szabályok
 - Megvalósítható alakzatok és viszonyaik
 - Elektromos szabályok (ERC)
 - Áramterhelhetőség
 - ESD, I/O
 - Szubsztrátkontaktusok, lebegő gate stb.
 - DFM szabályok
 - Gyárthatóság/ jó kihozatal
- ▶ LVS (layout vs. schematic)
 - Layout -> kapcsolási rajz + paraziták



Device



Interconnect

A design kit tartalma

- ▶ Tulajdonképpen minden konfigurációs és modell file, amire az eddig felsorolt eszközöknek szüksége van.
- ▶ Áramkör bevitel és szimuláció
 - A technológiában rendelkezésre álló alkatrészek
 - Szimbóluma
 - Áramköri modellje, különböző feltételek mellett. Ezek az ún. design corner-ek
 - Typical mean, Worst case power, Worst case speed, Worst noise
 - Különböző szimulátorokhoz
 - Analóg tervezés esetén a modell pontosságán múlik minden...
- ▶ Logikai szintézishez
 - A cellakönyvtár hdl leírása
 - A cellakönyvtár időzítései különböző feltételek mellett.
 - az ún. tlf (timing library format) file.

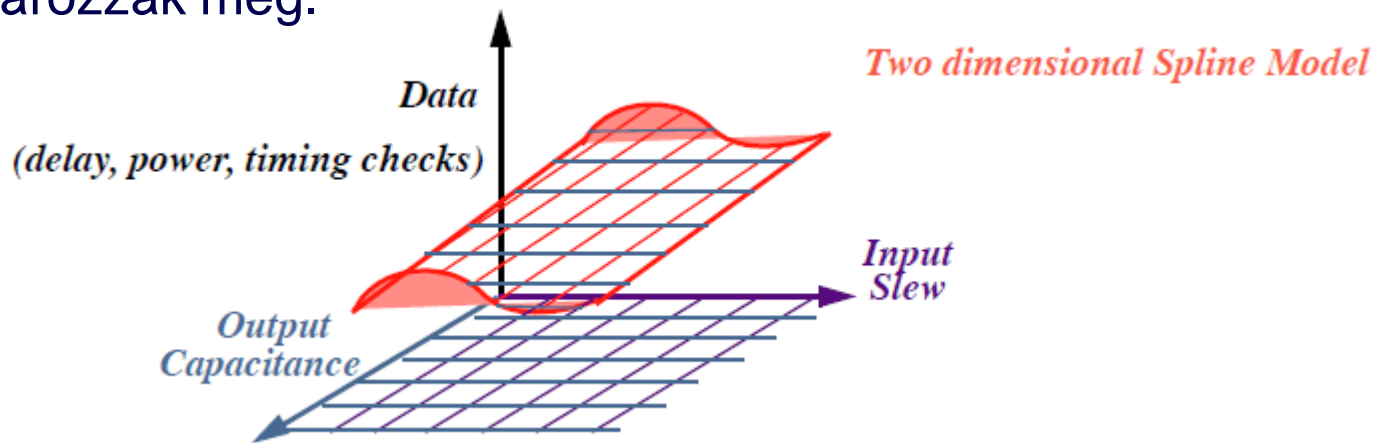
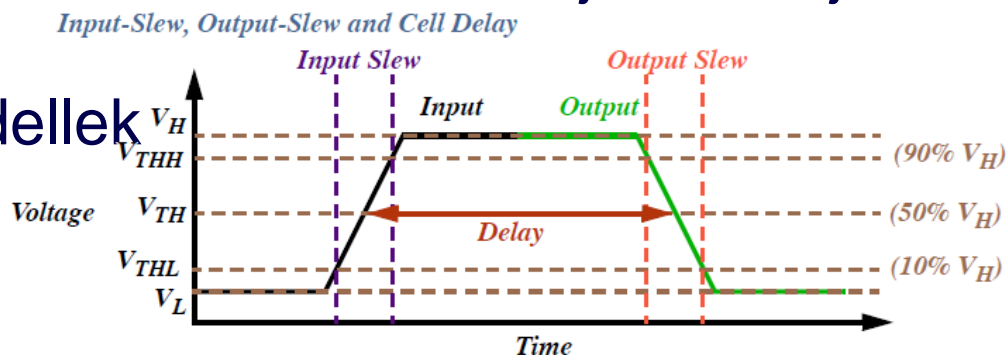


A TLF

▶ A standard cellakönyvtár áramköri szimulációjával állítják elő.

▶ Gyorsan számítható modellek

- Késleltetésre
- Fogyasztásra
- A kimeneti meredekségre
- A bemeneti jel meredeksége és a terhelés (wire load modellt alkalmaz, tehát kapacitás) függvényében.
- A kapu minden egyes jelútjára.
- Általában egy táblázat, a közbenső értékeket interpolációval határozzák meg.



A design kit tartalma

- ▶ A layout szintézishez
 - A rétegek leírása
 - A cellakönyvtár elemeinek layoutja (vagy layout absztraktja)
 - A huzalozási rétegek tulajdonságai
- ▶ Standard cellakönyvtár
 - Szimbólum
 - Kapcsolási rajz, visszafejtett parazitákkal
 - Layout
- ▶ Az ellenőrző eszközök számára
 - Tervezési szabályok különböző DRC programoknak
 - Visszafejtési szabályok



Tervezőrendszerek

▶ Főbb gyártók (abc sorrendben)

- Cadence Design Systems
- Mentor Graphics
- Synopsys
- Sok kisebb cég árul olyan eszközöket, amelyek beilleszthetők a nyílt tervezőrendszerekbe

▶ Főbb platformok

- (történelmi okokból) Unix
 - Solaris / HP UX / AIX
 - Enterprise Linux (RHEL, SLED)
- Windows

