

MOSFET kapcsolóeszközök VHDL szimulációja

Keresztes Péter

Széchenyi István Főiskola

A dolgozat egy VHDL csomag alapelveit ismerteti, ami lehetővé teszi a CMOS VLSI áramkörökben használt MOSFET eszközök modellezését. A modell a jelenlegi kivitelben egyirányú kapcsolóeszközök leírására alkalmas, de a lefedni kívánt alkalmazásokat tekintve ez nem korlátozó tényező. Ugyanakkor a csomag lehetővé teszi az átmeneti kapacitív szinttárolás és a gyenge logikai szintek modellezését. A bevezetett nyolc-állapotú bit az eddig alkalmazott többszintű típusoknál lényegesen egyszerűbb.

A dolgozat bemutatja egy digitális celluláris neurális hálózat processzor-chip (CASTLE) fejlesztésénél a VHDL csomaggal kapcsolatos tapasztalatokat.

1. Bevezetés

Az ún. FULL-CUSTOM CMOS VLSI áramkörök tervezése során megkerülhetetlen, hogy a logikai működést a tranzisztorok (kapcsolók) szintjén ellenőrizzük. A FULL-CUSTOM tervezés során ugyanis nehezen, vagy egyáltalán nem zárható le a cellák vagy makrocellák a LAYOUT tervezés során. A tervezők kénytelenek újra és újra módosítani egy-egy elemi funkciójú építőelem LAYOUT-ját, hiszen csak így lehet minimalizálni a chip területét.

Ez persze megköveteli, hogy a LAYOUT tervezéssel párhuzamosan kialakuló hierarchikus séma ne csak a LAYOUT-val való összehasonlítással történjen ellenőrzése, hanem logikailag is.

Ha a hierarchia bármely szintjén fenntartjuk az elrendezés megváltoztatásának a lehetőségét, akkor a „layout-versus-schematic” (LVS) ellenőrzés a fentiek miatt tranzisztor szinten kell hogy végbemenjen. Ez pedig csak akkor megnyugtató, ha a séma, vagyis a struktúrális leírás alapeleme a tranzisztor, azaz a szimuláció során használt viselkedési leírások tranzisztorok viselkedési leírásai.

CMOS áramkörök tranzisztor-szintű logikai szimulációja régóta kutatott terület. A MOSFET eszközöknek a logikai működés szempontjából van néhány meghatározó fontosságú tulajdonságuk. Ezek a következők:

- A MOSFET kétirányú kapcsoló. Az átvitel irányát a feszültség-viszonyok határozzák meg.
- A MOSFET vezérlő-elektrodája kapacitív, és töltéstároló tulajdonságát a CMOS kapcsolástechnikában alaposan ki is használják. (Dinamikus- és kvázistatikus-latch és flip-flop elemek)
- A MOSFET meghajtóképessége a csatornaellenállásától függ. Sok kapcsolás kihasználja ezt, és a MOSFET egy ellenállás szerepét tölti be. (Kvázi-n és kvázi-p CMOS kapuk).

- Az n-MOS a logikai magas szintet, a p-MOS a logikai alacsony szintet csak csonkítva képes átvinni. Ezeket a szintvesztéseket a két tranzisztor párhuzamos kapcsolásából álló átvivőkapu (transmission-gate) küszöböli ki. Ha ennek hibás vezérléseit a szimulációval ki akarjuk mutatni, modellezni kell a csonka szintű átvitteket.

Az eddig ezekre a kapcsolókra kifejlesztett VHDL, vagy más hardware-szimulációs eszközök túlhangsúlyozták a kétirányú (bipoláris) kapcsoló tulajdonságot, és az ellenállás-szerű meghajtást, ugyanakkor kevés gondot fordítottak a töltéstárolás és a csonka-szintű átvitel modellezésére [1], [2].

2. A nyolc-állapotú bit

A „newbit” típushoz nyolc állapotot ill. szintet rendelünk.

A szintek közül az '1' és a '0' a konvencionális, feszültség-generátoros meghajtású logikai magas és alacsony szintek.

A w0 és w1 ezeknek áramgenerátoros, ellenállásos meghajtású, illetve csökkentett feszültség-értékű megfelelőik.

CMOS áramkörökben előnyösen alkalmazhatók ún. kvázistatikus és dinamikus elemek, amelyekben a MOS kapacitás átmeneti töltéstároló tulajdonságát használják ki. Ezt a s0 és s1 tárolt szintekkel reprezentáljuk.

A hetedik állapotot, az u-t a tranziens állapot reprezentálására használjuk. Szemlélet kérdése, hogy az u szintet úgy tekintjük, mint a magas és az alacsony szintek közötti feszültség-állapothoz rendelt logikai szintet, vagy úgy, mint amelyik lehet éppen magas is, vagy lehet éppen alacsony is, de nem tudjuk, hogy ezek közül éppen milyen.

Végül a nyolcadik, a z szint a lebegő, ún. harmadik állapotot reprezentálja, tehát azt a szintet, amelyet egy kapcsoló akkor produkál, amikor elengedi a kimenetére kapcsolódó pontot.

Definiálunk egy speciális relációt. Ez a reláció a szintek közötti erőviszonyokat hivatott meghatározni. A relációt a következőképpen definiáljuk:

$level_i \text{---->} level_j$,

azaz $level_i$ felülírja a $level_j$ szintet, ha két azonos csomópontra kapcsolt kapcsolóeszköz közül - amelyek egyike $level_i$ -t, a másik $level_j$ -t ad a saját kimenetére - a $level_i$ -t képviselő szintje jelenik meg a közös csomóponton.

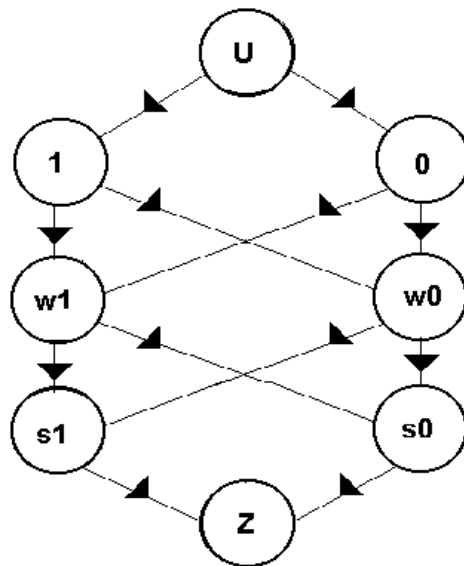
Ez a reláció *reflexív, antiszimmetrikus és tranzitív*, tehát a szintek halmazát részben rendezzi.

A részben rendezett szinteket mutatjuk be az 1. ábrán.

3. A „newbit” csomag felépítése

A „newbit” VHDL csomag definiálja a fent felsorolt szinteket, és a „connewbits” rezolúciós függvény segítségével érvényre juttatja a felülírási relációkat.

A függvény első részében található utasítássorozat segítségével számba vesszük a csomópontokra kapcsolódó összes meghajtót, mégpedig szintenként. A függvény második része a felülírási reláció alapján kiválasztja az érvényesülő szintet. A csomópont z állapotú csak akkor lehet, ha valamennyi meghajtó z kimenetű. Az u szint akkor jelenik meg, ha azonos erősségű, de egymásnak ellentmondó szintek vannak a meghajtó kimenetek között. Az erősebb, azaz a felülíró-szint mindig érvényesül a gyengébbel szemben.



1. ábra. A nyolc állapot, részben rendezve a felülírási reláció szerint

```
package newnstd is
type newbit is ( Z, w0, w1, s0, s1, '0', '1', U);
-- '0', '1' : strongest classical logical levels;
-- s0, s1 : stored logical levels;
-- w0, w1 : weak logical levels;
-- Z : floating (third) state;
-- U : undefined or transient state;
type newbit_vector is array (natural range<>) of newbit;
function connewbits (srcs : newbit_vector) return newbit;
end newnstd;
```

```
package body newnstd is
```

```
function connewbits ( srcs : newbit_vector) return newbit is
variable num0, num1, s0num, s1num, w0num, w1num, znum,
unum : natural := 0;
variable v : newbit := Z;
begin
for i in srcs'range loop
if srcs(i) = '0' then num0 := num0 + 1;
elsif srcs(i) = '1' then num1 := num1 + 1;
```

```

    elsif srcs(i) = s0 then s0num := s0num + 1;
    elsif srcs(i) = s1 then s1num := s1num + 1;
    elsif srcs(i) = w0 then w0num := w0num + 1;
    elsif srcs(i) = w1 then w1num := w1num + 1;
    elsif srcs(i) = Z then znum := znum + 1;
    else unum := unum + 1;
    end if;
end loop;

if unum > 0 then v := U; elsif
    unum = 0 and num0 = 0 and num1 > 0 then v := '1'; elsif
    unum = 0 and num1 = 0 and num0 > 0 then v := '0'; elsif
    unum = 0 and num1 > 0 and num0 > 0 then v := U; elsif
    unum = 0 and num1 = 0 and num0 = 0 and w0num > 0 and
        w1num = 0 then v := w0; elsif
    unum = 0 and num1 = 0 and num0 = 0 and w0num = 0 and
        w1num > 0 then v := w1; elsif
    unum = 0 and num1 = 0 and num0 = 0 and w0num = 0 and w1num = 0 and s1num = 0
        and s0num > 0 then v := s0; elsif
    unum = 0 and num1 = 0 and num0 = 0 and w0num = 0 and w1num = 0 and s1num > 0
        and s0num = 0 then v := s1; elsif
    unum = 0 and znum = srcs'length then v := Z;
else v:= Z;
end if;
return v;
end connewbits;
end newnstd;

```

4. A kapacitív szinttárolás modellezése a „newbit” csomag segítségével

Az alábbi VHDL leírás megmutatja, hogyan modellezhető egy, a MOSFET kapcsolók közé iktatott kapacitás. A CAPAC-elem, ha pontját **1**, **0**, **w1**, vagy **w0** szinttel meghajtja egy domináns kapcsoló, majd elengedi úgy, hogy azt a többi, erre a pontra irányuló kapcsoló is elengedi, ill. elengedve tartja, akkor a kapacitás a **z**-nél erősebb **s1** vagy **s0** szintet ad sajátmagára, és ezt egy megadott ideig (töltéstárolási idő) tartja, majd ő maga vált át a **z** szintre.

A töltéstárolásra képes kapu-elektrodájú MOSFET eszközök kapu-elektrodáját egy ezzel ekvivalens működésű utasítás-sorozattal modellezzük.

```

library work; use work.newnstd.all;
entity CAPAC is
    port
        ( CNODE : inout connewbits newbit);
end;

```

```

architecture BEH of CAPAC is
begin
    CNODE <= s1 when CNODE = '1' or CNODE = w1 else
        s0 when CNODE = '0' or CNODE = w0 else
        Z after 100 ns when CNODE = s1 or CNODE = s0 else
        Z;
end BEH;

```

5. MOSFET modellek

Példaként bemutatjuk az „erős”, **n**-csatornás MOSFET modellt. Ahogyan már említettük, a GATE elektróda kapacitásként viselkedik. A **S** (SOURCE) és a **D** (DRAIN) közötti kapcsolatot leíró programrész nem különbözteti meg, hogy a **G** elektródán **'1'**, **w1** vagy **s1**, illetve **'0'**, **w0**, vagy **s0** van.

Ugyanakkor a modell megkülönbözteti azt a két esetet, amikor az **S** elektródára **'0'** vagy **'1'** kapcsolódik. Fizikailag ugyanis a logikai magas szinttel vezérelt **n**-csatornás MOSFET a magas szintet csak küszöb-feszültséggel gyöngítve képes a **D** elektródán megjeleníteni. Ezért a modell ilyenkor **w1** szinttel jelentkezik. Ebben a modellben **w1** tehát a csökkentett feszültség-szintet reprezentálja.

```

library work; use work.newnstd.all;
entity nfet is
port
  ( S : in  connewbits newbit;
    D : inout connewbits newbit;
    G : inout connewbits newbit;
    nSUB : in connewbits newbit);
end;
architecture BEH of nfet is
begin
  G <= s1 when (G = '1' or G = w1) else
    s0 when (G = '0' or G = w0) else
    Z after 100 ns when G = s1 or G = s0 else
    Z;
  D <= S after 100 ps when (nSUB = '0' and (G = '1' or G = w1 or G = s1 or G = U) and
    (S = '0' or S = w0 or
    S = w1 or S = U or S = Z)) else
    w1 after 100 ps when (nSUB = '0' and (G = '1' or G = w1 or
    G = s1 or G = U) and S = '1') else
    s0 after 100 ps when (nsub = '0' and (D = '0' or D = w0) and
    (G = '0' or G = w0 or G = s0)) else

    s1 after 100 ps when (nsub = '0' and (D = '1' or D = w1) and
    (G = '0' or G = w0 or G = s0)) else
    Z after 10 ns when ((D = s0 or D = s1) and
    (G = '0' or G = w0 or G = s0) and nSUB = '0' )
    else
    Z after 100 ps when ( nsub = '0' and S = Z)
    else
    Z after 100 ps when (nsub = '0' and G = Z )
    else
    U after 100 ps when nSUB /= '0' else
    Z;
end BEH;

```

Egy másik **n**-csatornás **MOSFET** modellben az eszköz „gyenge”, azaz kimenetét az erős eszközhöz képest nagy ellenállású csatornán keresztül hajtja. Ezeket a szinteket is **w1** illetve **w0**-val jelöljük.

Álljon itt a kisáramú **n-MOSFET**, mint az ellenállásként alkalmazott **MOSFET** modellje.

```

library work; use work.newnstd.all;
entity R_nfet is
port
  ( S : in  connewbits newbit;
    D : inout connewbits newbit;
    G : inout connewbits newbit;
    nSUB : in connewbits newbit);
end;

architecture BEH of R_nfet is
begin
  G <= s1 when (G = '1' or G = w1) else
    s0 when (G = '0' or G = w0) else
    Z after 100 ns when G = s1 or G = s0 else
    Z;
  D <= w0 after 100 ps when nSUB = '0' and (G = '1' or G = w1 or G = s1 or G = U) and

```

```

S = '0' else
  w1 after 100 ps when nSUB = '0' and (G = '1' or G = w1 or G = s1 or G = U) and
    S = '1' else
  U after 100 ps when nSUB = '0' and (G = '1' or G = w1 or G = s1 or G = U) and
    (S = w0 or S = U) else
  U after 100 ps when nSUB = '0' and (G = '1' or G = w1 or G = s1 or G = U) and
    (S = w1 or S = U) else
  s0 after 100 ps when (nsub = '0' and (D = '0' or D = w0) and
    (G = '0' or G = w0 or G = s0)) else
  s1 after 100 ps when (nsub = '0' and (D = '1' or D = w1) and
    (G = '0' or G = w0 or G = s0)) else
  Z after 10 ns when ((D = s0 or D = s1) and
    (G = '0' or G = w0 or G = s0) and nSUB = '0') else
  Z after 100 ps when (nsub = '0' and S = Z) else
  Z after 100 ps when (nsub = '0' and G = Z) else
  U after 100 ps when nSUB /= '0' else
  Z;
end BEH;

```

6. A „newbit” csomag alkalmazása digitális CNN chip tervezése során

A Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató intézetének Roska Tamás akadémikus vezette analogikai kutató csoportja egy digitális CNN feldolgozást végző processzor architektúrát definiált [3]. A processzort, illetve egy processzor-tömböt tartalmazó chip fejlesztése a közelmúltban indult meg.

A tervezés során a **CADENCE „IC and System Design Package”** rendszerét használtuk. A tervezés egy **DATA-FLOW** stílusú, bit-vektor szintű **VHDL** leírás alapján, jellemzően „**bottom-up**” módon zajlott, egy szubmikronos **CMOS** technológiára. A minimális szilícium-felületre való törekvés jegyében a grafikus layout szerkesztővel megrajzolt elrendezés újra és újra módosításra került. Azért, hogy ezt az állandó layout módosulást a séma-leírásokkal követni tudjuk, az **LVS (Layout-Versus-Schematic)** programhoz előkészített strukturális leírás primitívjei az **n-mos** és **p-mos** tranzisztorok voltak.

Ez azt jelentette, hogy igen nagy és logikailag bonyolult cellákon *tranzisztor-szintű* szimulációt kellett végezni a tervezés ellenőrzése céljából. A tranzisztor-szintű **VHDL** leírást maga a **CADENCE** generálta, de a szimulációt a **V-SYSTEM (Model Technology)** rendszeren végeztük.

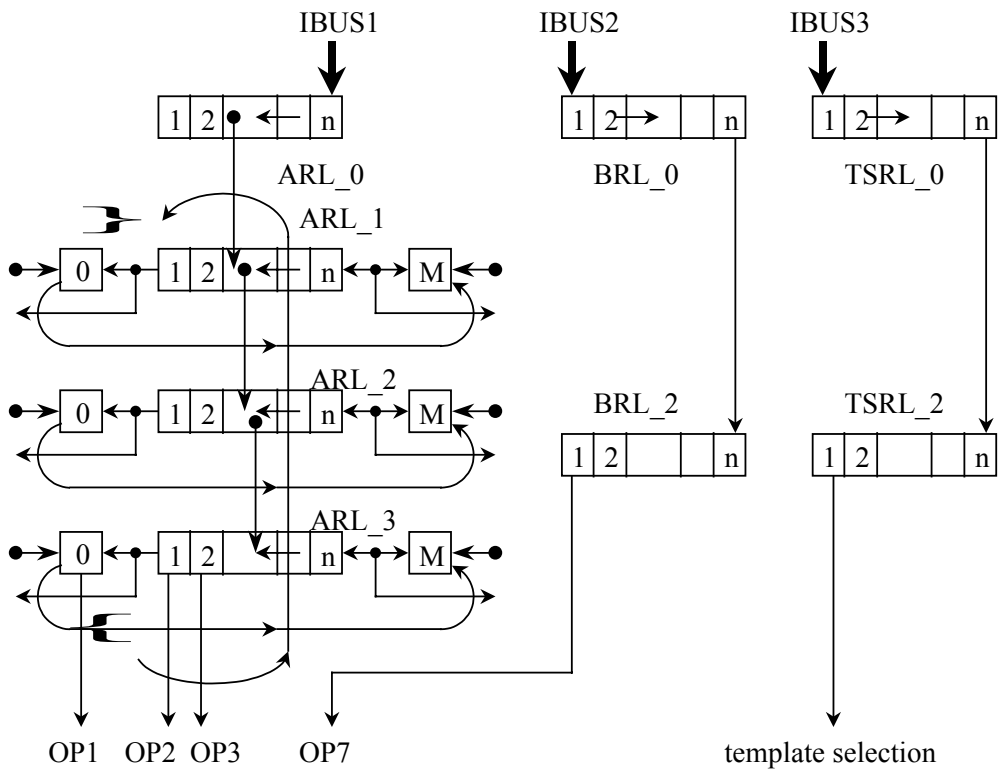
Mivel a **CASTLE**-chip időzítő-vezérlő egysége dinamikus-kétfázisú **M-S** tárolókból, regiszter-tömbje pedig kvázistatikus-dinamikus tárolókból áll, azaz a átmeneti töltéstárolás meghatározó szerepet játszik ezek működésében, az előző pontokban leírt **VHDL**-csomag és a tranzisztor modellek alkották az alapját a makrocella modellezésnek.

Számos tervezési hibát, mint a séma és a layout közötti eltérést maga az **LVS** rendszer mutatott ki. Számos olyan hiba is akadt, amely egyaránt megjelent a sémán és a layouton is, de csak a fenti csomagon alapuló tranzisztor-szintű szimuláció mutatta ki azokat.

Referenciák

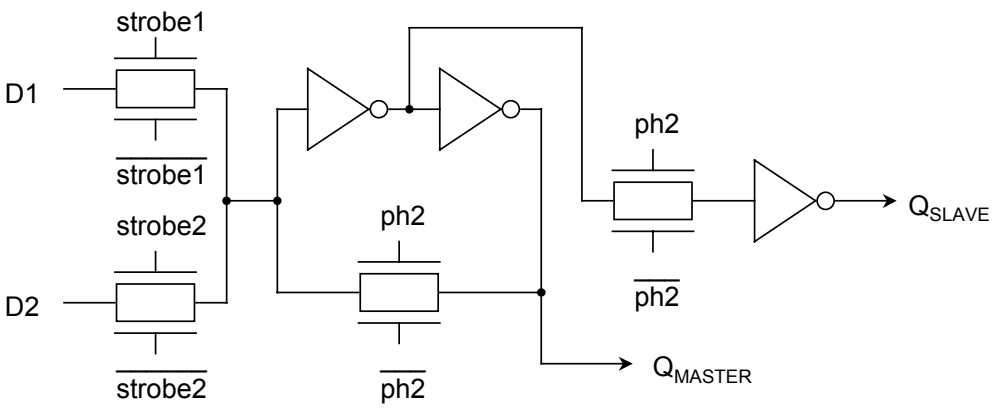
- [1.] A. Stanculescu: Bi-directional Switches in VHDL using the 46 Value System, VHDL Simulation Synthesis and Formal Proofs of Hardware, Ed. Jean Mermet, Kluwer Academic Publishers
- [2.] K, Khordoc, M. Biotteau, E. Cerny: Switch-Level Models in Multi-level VHDL Simulation, VHDL Simulation Synthesis and Formal Proofs of Hardware, Ed. Jean Mermet, Kluwer Academic Publishers
- [3.] P. Keresztes, Á. Zarándy, T. Roska, P. Szolgay, T. Bezák, T. Hidvégi, P. Jónás, A. Katona: An Emulated Digital CNN Implementation, Journal of VLSI Signal Processing Systems, Volume 23, Number 2/3, November/December 1999 pp 291-303. Kluwer Academic Publishers

(a 2. és 3. ábrákat lásd a következő oldalon)



2. ábra

A CASTLE processzor regiszter-tömbje. A regiszterek többsége több forrásból beírható, kvázistatikus MASTER, és dinamikus SLAVE fokozatból álló tároló-elemből áll.



3. ábra

Egy két forrásból beírható kvázistatikus-dinamikus tároló-elem. Az átmeneti tárolás az átvivőkapuk kimenetén, azaz ide kapcsolódó MOSFET vezérlőelektródákon történik.