

Логичко пројектовање

Предавање 1/12

**Увод у логичко
пројектовање**

VHDL 1/2

- Логичко пројектовање
 - Број логичких кола у мрежи велики – пројектовање ручно веома тешко.
 - Процес преласка са описа мреже на високом нивоу у облик који је спреман за имплементацију је јасан и дефинисан.
 - Computer-aided design (CAD) алати.
 - Hardware description languages (HDLs) – описују велике дигиталне системе без употребе шема.
 - Функционални опис система (таблица истинитости)
 - HDLs подржавају симулацију на различитим нивоима апстракције.
 - Top-down пројектовање.
 - Аутоматска синтеза.
 - Библиотеке и пакети.

VHDL 2/2

- VHDL (Very High-Speed Integration Circuit Hardware Description Language)
 - Данас су доминантни два језика за опис хардвера (VHDL и Verilog).
 - VHDL је акроним, Verilog није акроним већ име језика.
 - Употреба ова два језика скоро је једнако подељена у индустрији.
 - Када се научи један језик, једноставно је научити и други језик.
- У овом предмету је фокус на VHDL-у
 - VHDL се користи за учење концепата HDL-а.
 - VHDL је строжи у својој синтакси и конверзији променљивих него Verilog, па је то погоднија платформа за почетнике.
 - Основни принципи описа хардвера

Историја HDL-а 1/5

- Проналазак интегрисаног кола се приписују 2 патента из 1959. године.
 - Jack Colby, "Miniaturized Electronic Circuits", 1959, Texas Instruments.
 - Robert Noyce, "Semiconductor Device and Lead Structure", 1959, Fairchild Semiconductor.
 - Colby је 2000. године за свој изум добио Нобелову награду за физику, док је Noyce 1968. заједно са Gordon Moore-ом основао компанију *Intel*.
- Intel
 - 1971. године Intel је представио први микропроцесор са једним чипом користећи технологију интегрисаних кола, Intel 4004.
 - Овај микропроцесор је био интегрисано коло које је имало 2300 транзистора.
 - Ова серија микропроцесора лансирала је индустрију полупроводника, која је била покретачка снага раста Силицијумске долине, што је довело до 40 година невиђеног напретка технологије која је утицала на скоро сваки аспект савременог живора.

Историја HDL-а 2/5

- Moore-ов закон
 - Gordon Moore предвидео је 1965. године да ће се број транзистора у интегрисаном колу удвостручавати на сваке 2 године (Moore-ов закон)
 - Како је број транзистора у интегрисаном колу растао, тако је и расла комплексност пројектовања и функционалности које би се могле имплементирати.
 - Када је 1971. изумљен микропроцесор, повећане су могућности CAD алата што је омогућило постизање већих димензија пројектовања.
 - Ови већи и новији микропроцесори, омогућили су да CAD алати постану још већи и софистициранији и да за узврат дају још веће пројекте.
- Стандардизација
 - Брзо ширење дигиталних интегрисаних кола захтевало је од различитих произвођача међусобну компатибилност.
 - Да се документује сложено понашање већих система.
 - Употреба шема за документовање великих дигиталних пројеката постала је превише гламазна и тешко разумљива.
 - Описи речима је било лакше разумети, али чак је и овај облик документације постао превише обиман да би био ефикасан за велике пројекте.

Историја HDL-а 3/5

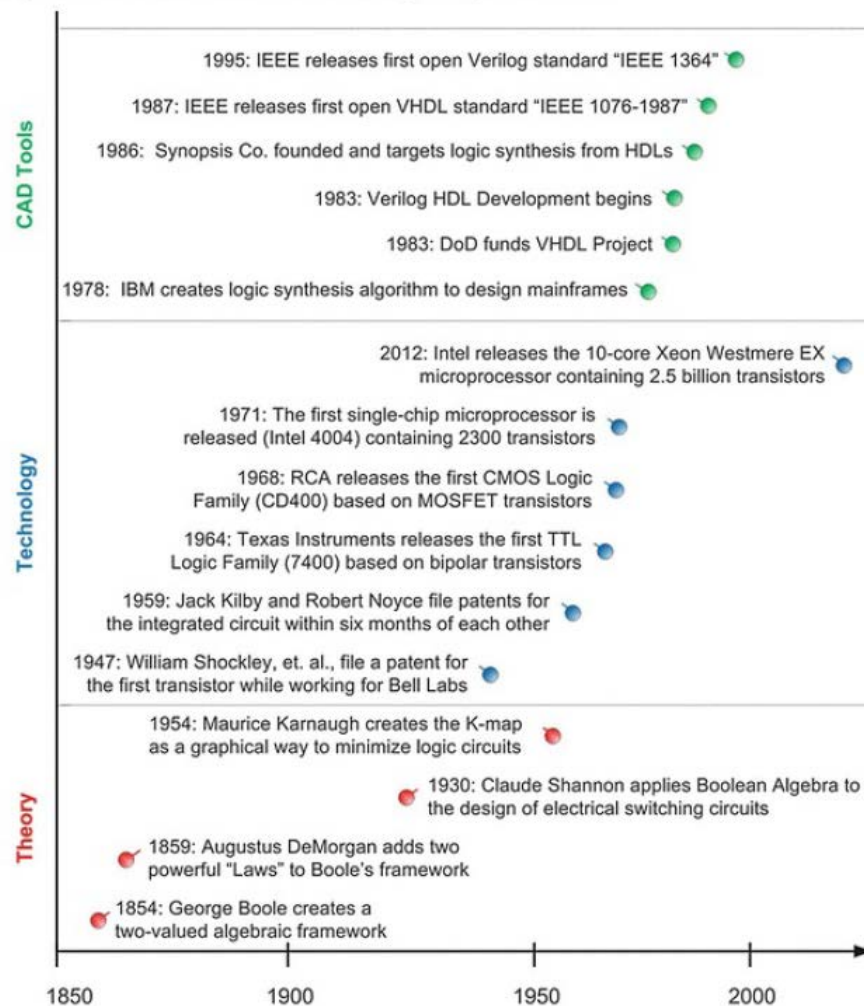
- ASIC
 - 1983, Министарство одбране САД (DoD) спонзорисало је програм за документовање понашања дигиталних система које би се могло користити код свих својих добављача.
 - Овај програм је мотивисан недостатком адекватних документација за функционалност интегрисаних кола специфичних за апликације (ASIC – Application Specific Integrated Circuits) који су били испоручени Министарству одбране.
 - Уз недостатак стандардизованог приступа документацији, добављачи су имали потешкоћа да репродукују еквивалентне делове онима који су постали застарели.
 - Министарство одбране је уговорило три компаније (Texas Instruments, IBM, Intermetrics) за развој стандардизованог алата за документацију који би обезбедио детаљне информације о интерфејсу (тј. о улазима и излазима) и о понашању дигиталних система.
- VHDL
 - Нови алат је требало да буде имплементиран у формату сличном програмском језику.
 - Могућност симулације морала је да обухвати више нивоа апстракције како би пружила максималну флексибилност.
 - 1985. објављена је прва верзија овог алата под називом VHDL.
 - Како би добили широко усвајање и обезбедили доследност употребе широм индустрије, VHDL је предат Институту за електротехнику, електронику и инжењерство (IEEE).
 - 1987, IEEE је објавио прву стандардну верзију VHDL-а.
 - Најновији VHDL стандард је IEEE-2008.

Историја HDL-а 4/5

- Verilog
 - 1983, Verilog HDL је развијен од стране компаније Automated Integrated Design Systems као језик за логичку симулацију.
 - Развој Verilog-а је текао потпуно независно од VHDL пројекта.
 - 1985, компанија Automated Integrated Design Systems је преименована у Gateway Design Automation, а потом 1990 у Cadence Design Systems.
 - Cadence је учинио Verilog отвореним за јавност како би остали конкурентни.
 - 1995, IEEE је издао Verilog стандард под називом IEEE-1364.
- Synopsis
 - Развој CAD алата за постизање аутоматизоване логичке синтезе датира из 1970-их када је IBM почео да развија серију алата за синтезу који су коришћени у дизајну њихових маинфреме рачунара.
 - 1986, главни напредак у синтези логике дошао је са оснивањем компаније под називом Synopsis.
 - Synopsis је била прва компанија на коју се фокусирала синтеза логике директно из HDL-а. Могућност симулације морала је да обухвати више нивоа апстракције како би пружила максималну флексибилност.
 - Ово је био велики допринос јер дизајнери су већ користили HDL-ове да опишу и симулирају свој дигитални систем, а сада је синтеза логике постала интегрисана у исти ток дизајна.
 - У почетку су били само нижи нивои апстракције који су темељно разрађени.
 - Како су се могућности CAD алата развијале, синтеза вишег нивос апстракције је постала могућа, али ни данас се не могу синтетизовати све функционалности које могу да се опишу у HDL-у.

Историја HDL-а 5/5

Major Milestones in the Advancement of Digital Logic and HDLs



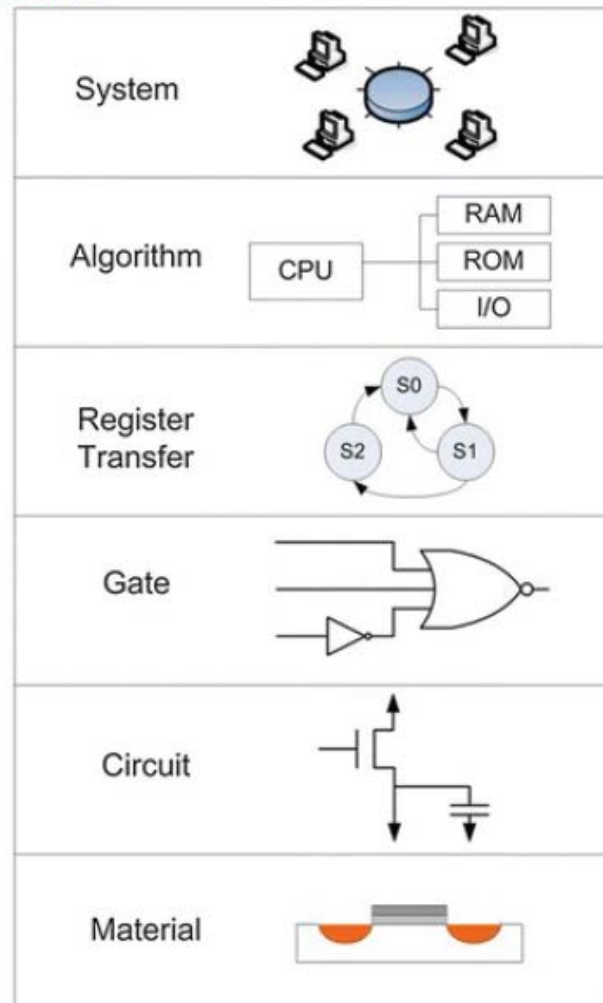
HDL апстракција 1/7

- HDL-ови су првобитно дефинисани да би могли да моделују понашање на више нивоа апстракције.
- Апстракција је важан концепт у инжењерству јер се омогућава да опишемо како ће системи функционисати где се не троши време на детаље имплементације.
- Такође, уклањањем детаља имплементације на нижим нивоима, симулације за моделовање функционалности вишег нивоа се могу извршавати у реалном времену.
- Када би се комплетан компјутерски систем симулирао коришћењем детаљних модела за сваки MOSFET, било би потребно огромно време да се изврши ова симулација.
- Графички приказ различитих слојева апстракције у дизајну дигиталног система је приказан на следећој слици.
-
-
-

HDL апстракција 2/7

Levels of Design Abstraction

Abstraction ↑



↓ Low-Level Detail

HDL апстракција 3/7

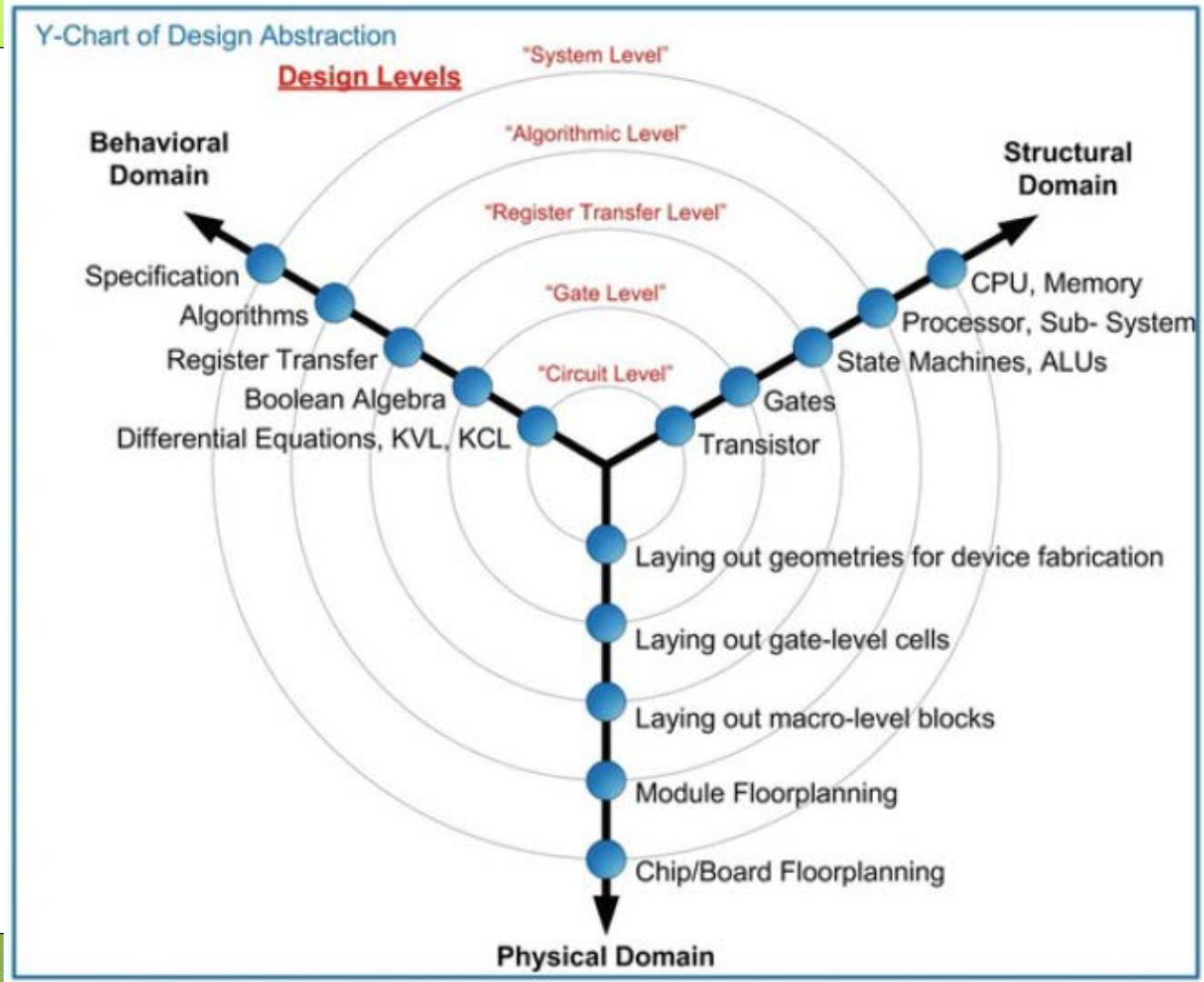
- Највиши ниво апстракције је системски ниво. На овом нивоу понашање система је описано навођењем скупа широких спецификација. Пример дизајна на овом нивоу је спецификација као што је „рачунарски систем ће извести 10 Tera Floating Point Operations per Second (10 TFLOPS) са дуплом прецизношћу и неће потрошити више од 100 W енергије”. Примећује се да ова спецификација не диктира детаље нижег нивоа као што је тип логике или тип рачунарске архитектуре која се користи.
- Један ниво ниже од системског нивоа је алгоритамски ниво. На овом нивоу спецификације почињу да се разбијају на подсистеме, од којих сваки има повезано понашање које ће остварити део примарног задатка. На овом нивоу, пример спецификације рачунара се може поделити на подсистеме као што је централна процесорска јединица (CPU) за обављање рачунања и меморија са случајним приступом (RAM) за чување улаза и излаза рачунања.
- Један ниво ниже од алгоритамског нивоа је ниво преноса регистра (RTL). На овом нивоу су описани детаљи како се подаци премештају између и унутар подсистема и како се манипулише подацима на основу системских улаза.
- Један ниво ниже од RTL нивоа је gate ниво. На овом нивоу, дизајн је описан коришћењем основних gate-ова и регистара. Gate ниво је у суштини шематски (било графички или текстуални) који садржи компоненте и везе које ће имплементирати функционалности са горњих нивоа апстракције.

HDL апстракција 4/7

- Један ниво ниже од gate нивоа је ниво кола. Ниво кола описује рад основних gate-ова и регистара помоћу транзистора, жица и других електричних компоненти као што су отпорници и кондензатори.
- Коначно, најнижи ниво апстракције дизајна је материјални ниво. Овај ниво описује колико се различитих материјала комбинују и обликују како би се имплементирали транзистори уређаја и везе на нивоу логичких кола.
-
- HDL-ови су дизајнирани да моделују понашање на свим овим нивоима уз изузетак материјалног нивоа.
- Иако постоји одређена способност за моделовање нивоа кола, понашања као што су MOSFET-ови као идеални прекидачи и pull-up/pull-down отпорници, HDL се обично не користе на нивоу кола.
- Други графички приказ апстракције дизајна познат је као Gajski и Kuhn's Y-графикон. Y-графикон приказује апстракцију у три различита домена дизајна: понашање, структурни и физички. Сваки од ових домена дизајна садржи нивое апстракције (тј. систем, алгоритам, RTL, gate и коло).
- Пример Y-графикон-а је приказан на следећој слици.
-

HDL апстракција

5/7



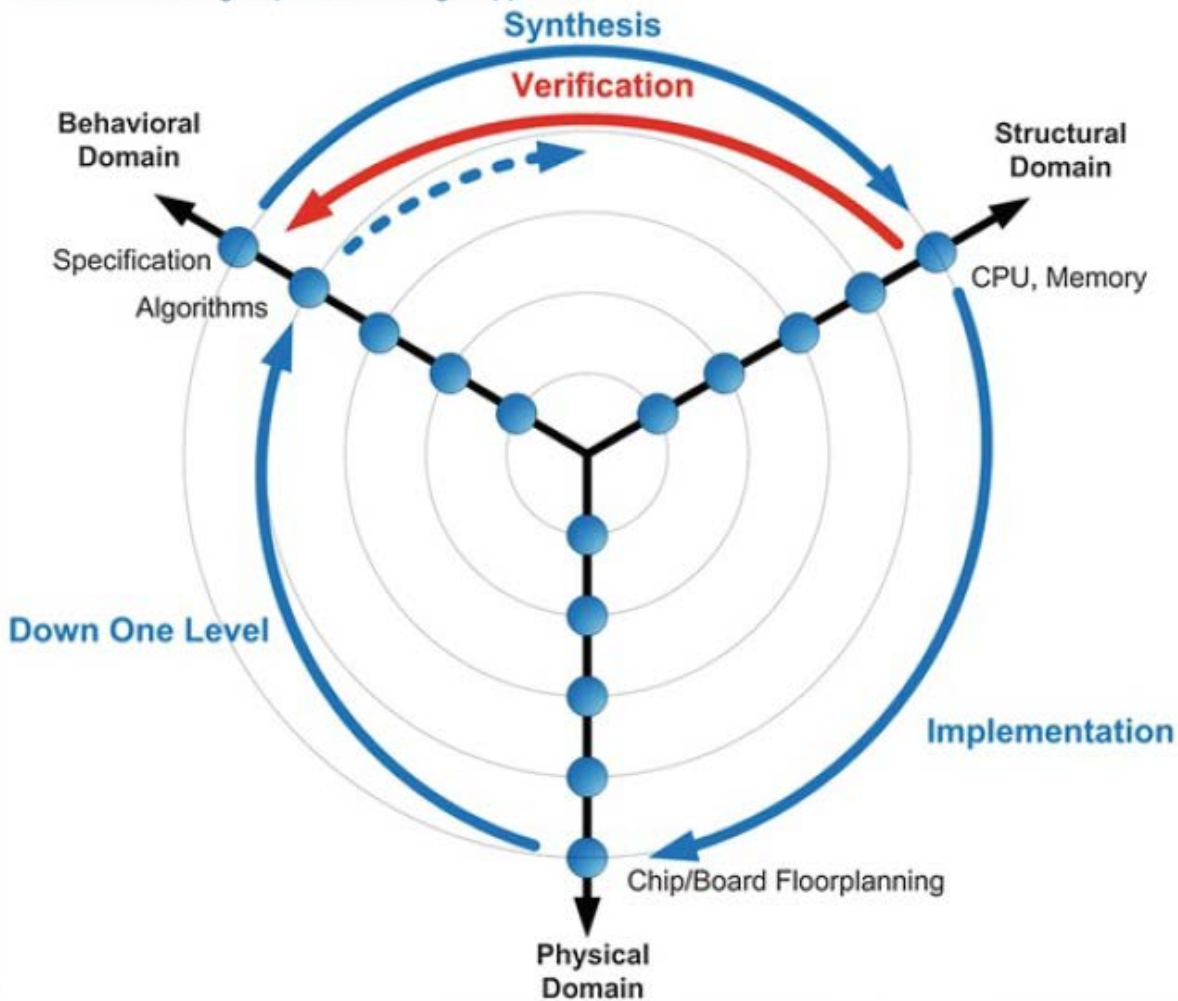
HDL апстракција 6/7

- Y-графикон такође приказује како су нивои апстракције различитих домена дизајна међусобно повезани.
- Ток дизајна одозго надоле може се визуелизовати у Y-графикону и спиралом према унутра у смеру казаљке на сату. Прелазак са домена понашања на структурни домен је процес синтезе.
- Кад год се синтеза изводи, резултујући систем треба упоредити са претходним описом понашања. Ова провера се зове верификација. Процес стварања физичких кола која одговарају структурном опису се назива имплементација.
- Спирала се наставља доле кроз нивое апстракције све док се дизајн не реализује на нивоу геометрије која представља елементе логичких кола (транзистори, везе, итд.) који су спремни за производњу у силицијуму.
- Процес дизајна одозго надоле приказана као унутрашња спирала на Y-графикону је приказана на следећој слици.
-
- Y-графикон представља формални приступ за велике дигиталне системе. За велике система који су дизајнирани од стране тимова инжењера, кључно је да се формално, одозго-надоле прати процес пројектовања како би се елиминисале потенцијално скупе грешке у пројектовању као и имплементације која се спроводи на нижим нивоима апстракције.

HDL апстракција

7/7

Y-Chart Illustrating Top-Down Design Approach



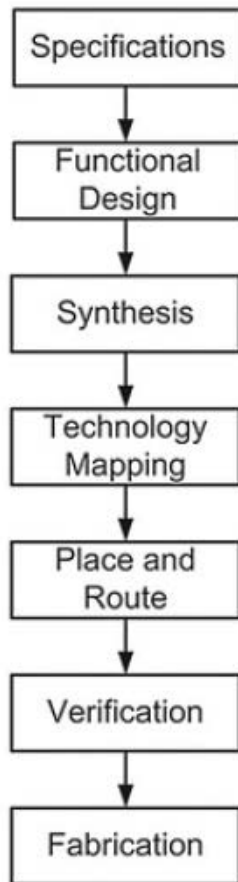
Ток модерног дигиталног дизајна 1/5

- Приликом извођења мањег дизајна или дизајна подсистема, процес се може поделити на појединачне кораке. Ови кораци су приказани на следећој слици.
- Овај процес је дат генерално и примењује се и на класичан и модеран дигитални дизајн.
- Разлика између класичног и модерног је у томе да модеран дигитални дизајн користи HDL и аутоматизоване CAD алате за симулацију, синтезу, рутирање и верификацију.
-
-

Ток модерног дигиталног дизајна 2/5

Digital Design Flow

Steps



Description of Tasks at Each Step

- State the desired behavior of the design using broad, high-level specifications.
- Describe the high-level architecture of the design (e.g., block diagrams for inputs/outputs, sub-systems) and generic behavior (truth tables, state diagrams and/or algorithms).
- Create the gate-level connection (schematic or netlist) of the design using logic synthesis processes (e.g., K-maps or automated CAD tools).
- Select the logic technology that will achieve the specifications (e.g., 74HC family, 32nm CMOS ASIC). Manipulate the gate-level netlist/schematic into a form that is suitable for this technology (e.g., DeMorgan's NAND/NOR).
- Arrange the components to minimize the area needed (on a board or chip) and wire all connections to minimize interconnect length and crossings.
- Once a technology is chosen and the routing is complete, the gate and wiring delays can be used to estimate whether the final design meets the timing and power consumption requirements of the original specifications.
- Once the design is verified it can be implemented. (ASIC, programmable device, board-level, discrete parts)

Ток модерног дигиталног дизајна 3/5

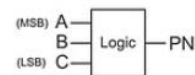
- Овај генерички ток процеса дизајна може се користити у класичном и модерном дигиталном дизајну, иако савремени дигитални дизајн омогућава додатну верификацију на сваком кораку користећи аутоматизоване CAD алате.
- Следећа слика показује како се овај ток користи у класичном приступу дизајну комбинационог логичког кола.
-
- Модеран ток дизајна заснован на HDL-овима укључује могућност симулације функционалности на сваком кораку процеса. Функционалне симулације могу бити извршене на почетном опису понашања система. На сваком кораку процеса дизајна, функционалност је описана детаљније, на крају се креће ка кораку израде.
- На сваком нивоу, детаљне информације могу бити укључене у симулацију да би се проверило да ли је функционалност и даље исправна и да ли дизајн и даље испуњава оригиналне спецификације. Друга следећа слика приказује модеран дигитални ток дизајна са укључивањем могућности симулације на сваком кораку.

Ток модерног дигиталног дизајна 4/5

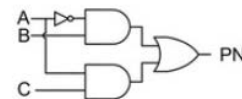
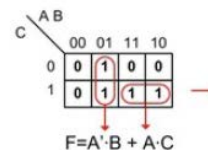
Classical Digital Design Flow



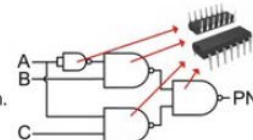
- Design a "Prime Number Detector" that takes in values from 0_{10} to 7_{10} . The circuit should be able to indicate a prime number with a delay less than 200ns.



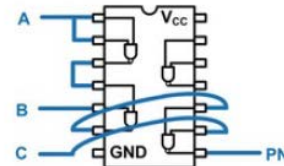
A	B	C	PN
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



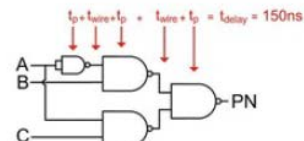
- It is decided that a 74HC logic family will be the most cost-effective technology for this design. To minimize the number of parts, the logic will be implemented with only NAND-gates.



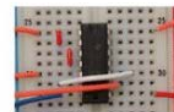
- The circuit to be implemented is placed in a floor plan and an estimate of the connections are made.



- Based on the layout, the wire delays are found. The delays of the gates are taken from the data sheet.

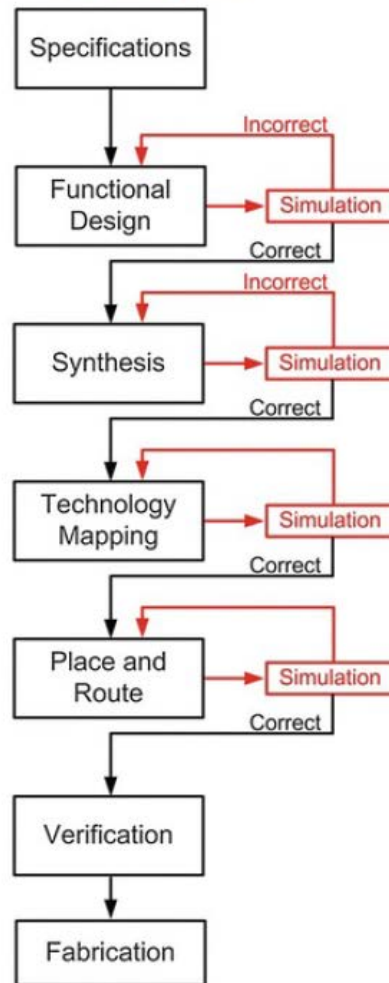


- The verified circuit is implemented in hardware.



Ток модерног дигиталног дизајна 5/5

Modern Digital Design Flow



- The initial design is in the form of an HDL behavioral description. This design is simulated to verify its proper functionality.

- After synthesis, the design is described at the gate-level. A logic simulation is used to verify that the functionality of the gate-level logic matches the functionality of the pre-synthesis behavioral description.

- After technology mapping, an estimate of the gate delays can be used in the simulation to make sure the timing requirements of the design are met.

- After place and route, an estimate of the wiring delays can be included in the simulation to make sure the timing requirements of the design are met.

- The final design is analyzed to see if it meets the original design specifications.

- Fabrication is typically in the form of an ASIC or a programmable device.