

2 ZÁKLADNÉ LOGICKÉ HRADLÁ NOR, OR, AND, NAND

Projekt ukazuje postup jednoduchého návrhu „krok za krokom“ vo vývojom systéme Quartus II – vytvorenie nového projektu, nových súborov návrhu, kompiláciu, simuláciu a programovanie. Projekt je vytvorený vo vývojom systéme Quartus II ver.4.2.

2.1 ZADANIE PRÍKLADU Č.1

Vo vývojovom prostredí Quartus II realizujte projekt na rozsvietenie desatinnej bodky MSD (MSD_dp) sedem segmentovej LED stlačením jedného z tlačidiel (PB1, PB2) na vývojovej doske UP1 CPLD. Návrh realizujte v grafickom editore. Čísla použitých pinov a ich funkcie sú uvedené v tabuľke č.1.

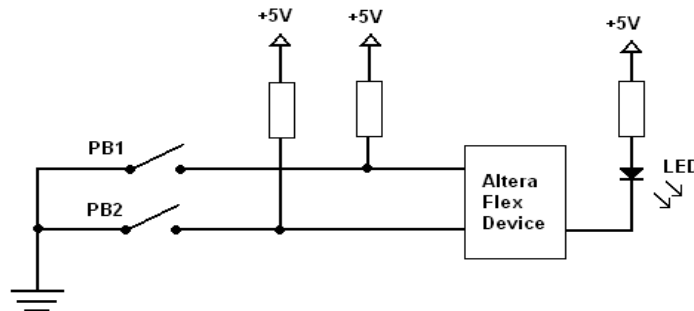
Meno pinu	Typ pinu	Pin	Funkcia pinu
PB1	Vstup	28	Tlačidlo 1 (0 = stlačené tlačidlo)
PB2	Vstup	29	Tlačidlo 2 (0 = stlačené tlačidlo)
MSD_dp	Výstup	14	Desatinná bodka sedem segmentovej LED (0 = LED ON = svieti, 1 = LED OFF= nesvieti)

Tab.1: Tabuľka pinov

2.2 RIEŠENIE PRÍKLADU Č.1

Úlohu môžeme riešiť návrhom logického obvodu, ktorý zmení logickú hodnotu pinu na ktorý je pripojená desatinná bodka MSD sedem segmentovej LED pri stlačení ľubovoľného tlačidla, PB1 alebo PB2. Takýto logický obvod môžeme realizovať použitím základného logického hradla NOR, ktorý má dva vstupy (PB1 a PB2) a jeden

výstup (MSD_dp). Tlačidlá PB1, PB2 a segment LED MSD_dp na vývojovej doske UP1 sú zapojené podľa obr.1.

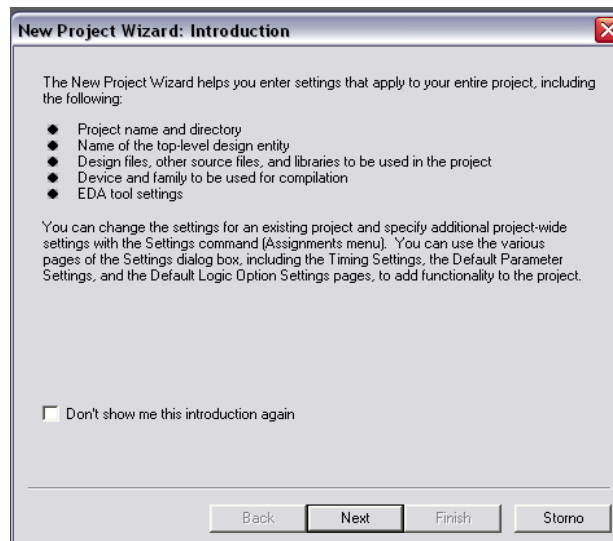


Obr.1: Schéma zapojenia PB1, PB2 a LED na vývojovej doske UP1 CPLD


2.3 POSTUP RIEŠENIA VO VÝVOJOM PROSTREDÍ QUARTUS II

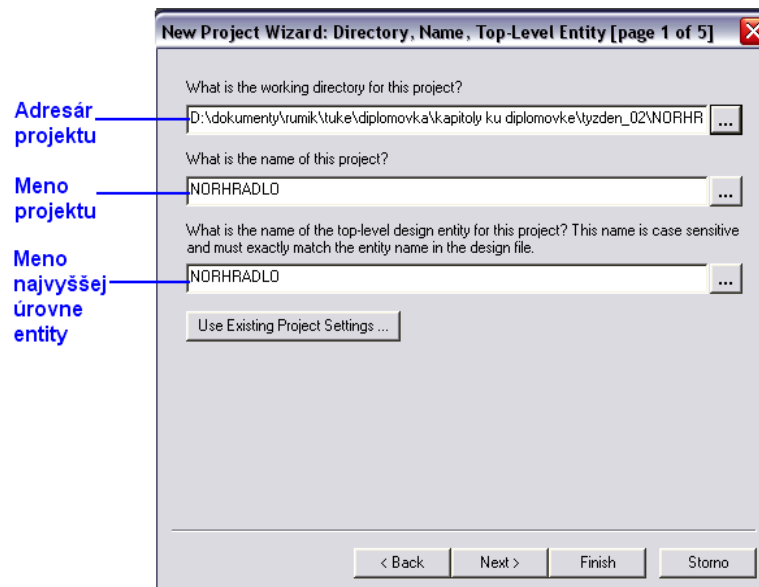
2.3.1 VYTVORENIE NOVÉHO PROJEKTU

Nový projekt vytvoríme voľbou **New Project Wizard** z **File** menu, po ktorej sa objaví úvodné okno (obr.2). Kliknutím na tlačidlo **Next** sa objaví sa prvé okno (**New Projekt Wizard**), kde sa definuje miesto uloženia a názov projektu a entity (obr.3).



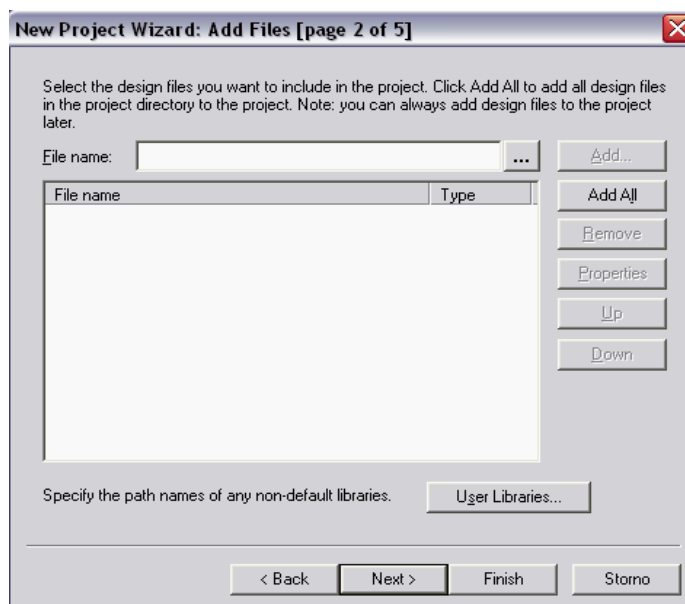
Obr.2: Úvodné okno pri vytváraní nového projektu vo vývojovom prostredí Quartus II

Kliknutím na tlačidlo  prvého riadku, sa objaví okno, kde si môžeme vybrať cestu k adresáru, do ktorého uložíme projekt a do ktorého budú ukladané všetky vytvárané súbory, alebo si môžeme vytvoriť nový adresár. Pre tento projekt si vytvoríme adresár *NORHRADLO*. V druhom riadku definujeme názov projektu a v treťom názov najvyššej úrovne entity. Názov projektu a entity môže, ale nemusí, byť taký istý ako názov adresára do ktorého budeme ukladať projekt (väčšinou sa volia rovnaké). Pre tento projekt definujeme rovnaký názov projektu aj názov entity ako je názov adresára – *norhradlo*. Ak sme v prvom okne všetko zadefinovali klikneme na tlačidlo **Next**.



*Obr.3: Prvé okno voľby **New Projekt Wizard** – okno definície miesta, najvyššej úrovne a názvu entity*

V nasledujúcom okne priradzujeme súbory do projektu. Napríklad, ak v nejakom inom projekte je vytvorený VHDL kód, ktorý budeme používať aj v novom projekte, tak si ho tu môžeme pridať (obr.4). V tomto projekte nebudeme pridávať žiaden súbor – klikneme na **Next**.



Obr.4: Druhé okno voľby **New Project Wizard** – okno, priradenia súborov do projektu

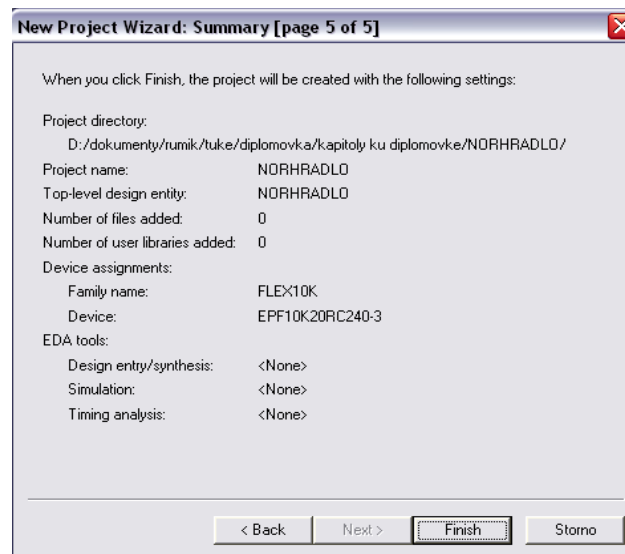
Objaví sa okno – tretie okno voľby **New project Wizard**, v ktorom si definujeme rodinu obvodov, v našom prípade rodinu FLEX10K.

V časti **Target device** zaškrtneme možnosť *Specific devices selected in 'Available devices' list*.

V spodnom okne vyberieme presný typ obvodu, s ktorým chceme pracovať. V našom prípade súčiastku EPF10K20RC240-4. Vo web verzii vývojového prostredia Quartus II ver.4.2 sa súčiastka EPF10K20RC240-4 nemusí nachádzať. Potom zvolíme súčiastku EPF10K20RC240-3. Rozdiel je len v rýchlosti logiky súčiastok.

Vo štvrtom okne **New project Wizard** si môžeme nastaviť nástroje EDA. Tu nebudeme nič nastavovať.

V poslednom okne - piate okno voľby **New Project Wizard**, sú zobrazené všetky voľby a nastavenia, ktoré sme v predchádzajúcich krokoch vykonali. Ak je všetko v súlade s našimi požiadavkami, klikneme na tlačidlo **Finish**. Ak niektoré nastavenie nevyhovuje, vrátime sa k príslušnej voľbe nastavenia tlačidlom **Back**. (obr.5)



Obr.5: Posledné okno voľby *New Projekt Wizard* – zobrazuje všetky nastavenia

Po potvrdení všetkých nastavení v **New Project Wizard**, Quartus II vytvorí tieto súbory:

release.fsf – súbor, v ktorom sú definované nastavenia softvéru

debug.fsf – súbor, v ktorom sú tiež nastavenia pre softvér

norhradlo.ssf – súbor, v ktorom sú definované nastavenia simulátora

norhradlo.csf – súbor, obsahujúci nastavenia kompilátora

norhradlo.psf – súbor, ktorý obsahuje preddefinované nastavenia pomocníka pri návrhu (**Design Assistant**)

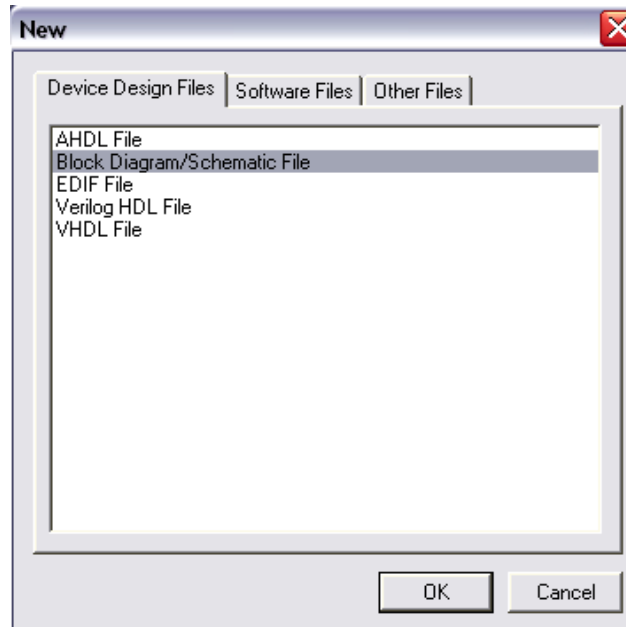
norhradlo.qpf – súbor, ktorý reprezentuje projekt

Okrem týchto súborov sa vygeneruje aj adresár *db*, do ktorého sa ukladajú pomocné súbory.

2.3.2 VYTVORENIE NÁVRHU SCHÉMATICKÝM/GRAFICKÝM EDITOROM (SÚBOR *.BDF*)

Tento súbor vytvoríme vykonaním nasledujúcich krokov:

- Vyberieme položku **New** z **File menu**
- V záložke **Design Files** si zvolíme možnosť **Block Diagram/Schematic File** (obr.6)



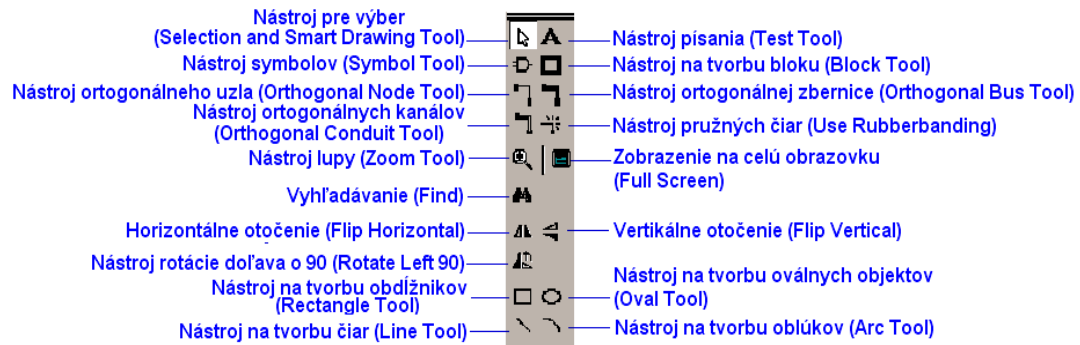
Obr.6: Výber možnosti **Block Diagram/Schematic File**

- Kliknutím na **OK** sa otvorí okno blokového editora
- Vyberieme položku **Save As** z **File menu**
- Vyberieme adresár *NORHRADLO* a uložíme náš súbor, ktorý nazveme *norhradlo.bdf*. Pod riadkom, kde sa definuje názov projektu zaškrtneme **Add file to current project** (pridať súbor do aktuálneho projektu).
- Potvrdíme tlačidlom **Save**. Tým sme uložili aj vložili súbor do projektu.

2.3.3 VYTVORENIE (NAKRESLENIE) SCHÉMY

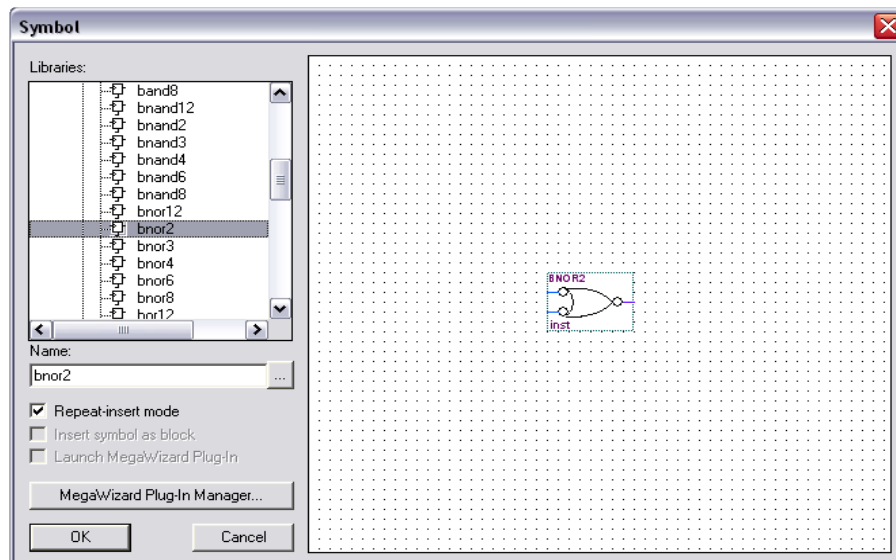
Schému vytvoríme nasledujúcim postupom:

- Na panely nástrojov (obr.7) klikneme na **Symbol Tool**



Obr.7: Panel nástrojov

- V okne **Libraries**, ktoré sa objaví vyberieme `c:\quartus\libraries\→primitives→logic`
- Z adresára `logic` vyberieme symbol **BNOR2** a potvrdíme stlačením **OK** (obr.8)



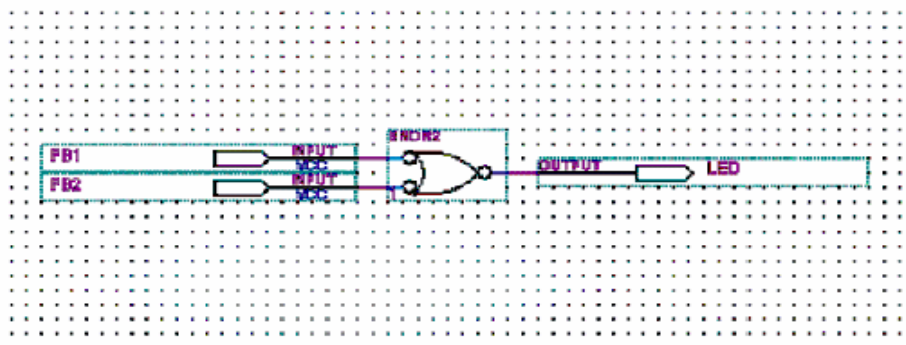
Obr.8: Výber prvku

- Symbol umiestnime na požadované miesto pohybom myšky a vložíme ho kliknutím ľavého tlačidla
- Klikneme na **Symbol Tool**

- Z adresára **pin** vyberieme a umiestnime na požadované miesto symbol **input** (vstup) dvakrát a potom symbol **output** (výstup)

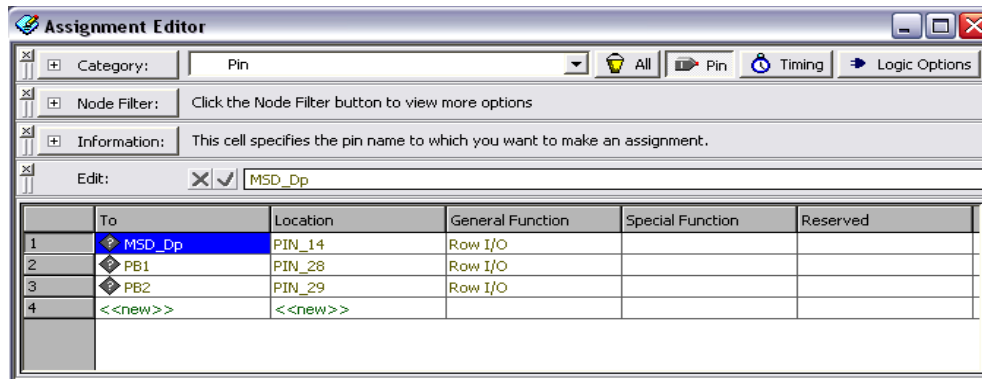
Jednotlivé vývody prepojíme vodičmi tak, že klikneme na **Orthogonal Node Tool**, priblížime sa s myšou k vývodu (vstupu alebo výstupu) symbolu a ak sa kurzor myši zmení na krížik klikneme a držíme ľavé tlačidlo myši a zároveň sa pohybujeme k vývodu ktorý chceme spojiť.

Názov a hodnotu priradíme k pinom tak, že klikneme na pin pravým tlačidlom myši, zobrazí sa tabuľka, z ktorej vyberieme položku **Properties**. Zobrazí sa nám okno v ktorom môžeme zadefinovať názov príslušného pinu, v riadku **pin name(s)** zadefinujeme názov pinu (PB1, PB2, MSD_dp) a v riadku **Default value** hodnotu vstupného pinu (VCC). (Výsledná schéma je znázornená na obr.9)



Obr.9: Výsledná schéma návrhu v blokovom editore

Pinom priradíme ich príslúchajúce čísla príkazom **Pins** v **Assignments menu** (obr.10). Do riadku z názvom **Edit**, v editovacom okne, napíšeme názov pinu (**PB1**). Potvrdíme Entrom, alebo stlačením tlačidla so zelenou fajkou. Tento názov pinu sa zobrazí v okne pod editovacím oknom. V príslušnom riadku dvakrát klikneme na kolónku v stĺpci s názvom **Location**, vyberieme číslo pinu a opäť potvrdíme Entrom.



Obr.10: Priradenie pinov

V prípade pinov PB2 a MSD_dp postupujeme rovnako ako pri pine PB1.

Zadefinovaním všetkých pinov okno zavrieme a zmeny uložíme.

2.3.4 KOMPILÁCIA

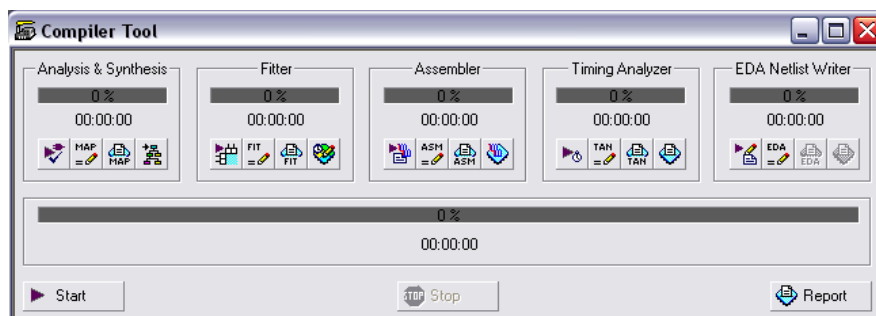
Kompilátor v softvéri Quartus II obsahuje niekoľko modulov, ktoré môžeme využiť pri práci s návrhom. Ide o moduly:

- Analysis & Synthesis (Analýza a syntéza)
- Fitter (nástroj na rozmiestnenie a prepojovanie)
- Assembler (Asemblér)
- Timing Analyzer (Časová analýza)

V Quartus II sa môžu moduly spustiť naraz, alebo voľbou **Processing menu** → **Start** môže spúšťať jednotlivé moduly aj samostatne. (napr.: **Processing menu** → **Start** → **Start Fitting**).

Spustenie kompilácie

Kompiláciu môžeme spustiť voľbou **Start Compilation** z **Processing menu**, pomocou ikony na panely nástrojov, alebo prostredníctvom príkazu **Compiler Tool** z **Tools** menu (obr.11).



Obr.11: Prostriedky kompilácie

Po spustení kompilácie sa automaticky zobrazia tieto okná :

- **Status Window** (obr.12): v tomto okne môžeme sledovať priebeh kompilácie. Môžeme pozorovať čas, za ktorý jednotlivé moduly spracovali projekt, aj celkový čas kompilácie. Okrem toho je tu znázornená aj miera úspešnosti celkovej kompilácie a jednotlivých modulov v percentách.

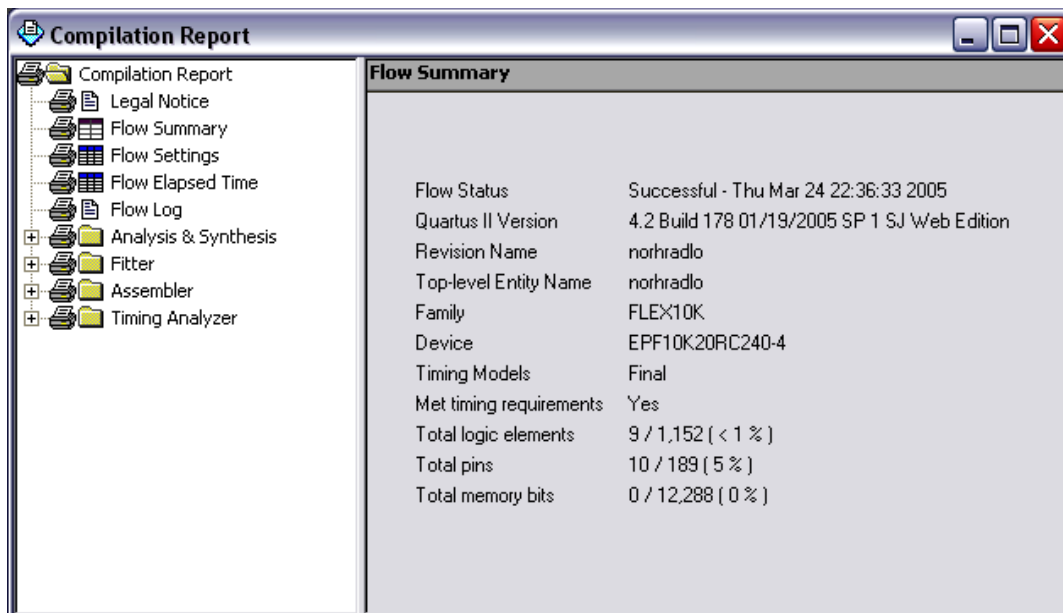
Module	Progress %	Time
Full Compilation	84 %	00:02:43
Analysis & Synthesis	100 %	00:00:31
Fitter	100 %	00:01:12
Assembler	100 %	00:00:54
Timing Analyzer	38 %	00:00:06

Čas, za ktorý sa vykonala celá kompilácia

Čas spracovania jednotlivých modulov kompilácie

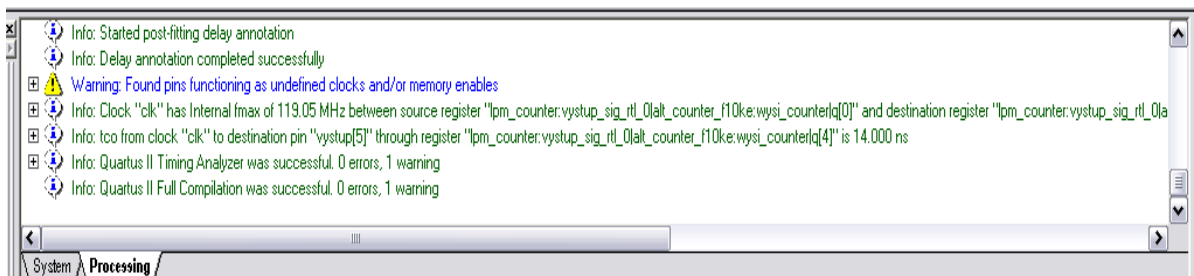
Obr.12: Stav kompilácie

- Norhľadlo **Compilation Report** (obr.13): Okno je rozdelené na dve časti: ľavé okno a pravé okno. V ľavom okne sú uvedené výsledky kompilácie, usporiadané do adresárov, v stromovej štruktúre, ktoré obsahujú súbory podľa toho, ktorým modulom boli generované. V pravom okne sú zobrazené najdôležitejšie výsledky kompilácie.



Obr.13: Okno výsledkov kompilácie

- **Message Window** (obr.14): V tomto okne sú zobrazované všetky správy, ktoré sú počas kompilácie generované. Zobrazené správy môžu byť: informačné, varovné a chybové. Ak sa objavia chybové správy, Quartus II umožňuje lokalizovať túto chybu priamo v súbore návrhu (dvojklik ľavým tlačidlom myši na správu). Okrem toho ponúka aj pomoc pri odstránení chyby (kliknutím na správu pravým tlačidlom sa zobrazí tabuľka, v ktorej vyberieme položku **Help**).



Obr.14: Okno správ

Po úspešnej kompilácii môžeme projekt simulovať a konfigurovať.

Súbory, ktoré vznikajú pri kompilácii sú:

norhradlo.map.rpt – súbor správ generovaný modulom analýzy a syntézy

norhradlo.fit.rpt – súbor správ generovaný modulom na rozmiestnenie
a prepojovanie

norhradlo.asm.rpt – súbor správ generovaný asemblérom

norhradlo.tan.rpt – súbor správ generovaný časovým analyzátorom

norhradlo.map.eqn – súbor operácií (výpočtov), ktoré vykonáva modul analýzy
a syntézy

norhradlor.fit.eqn – súbor operácií (výpočtov), ktoré vykonáva fitter

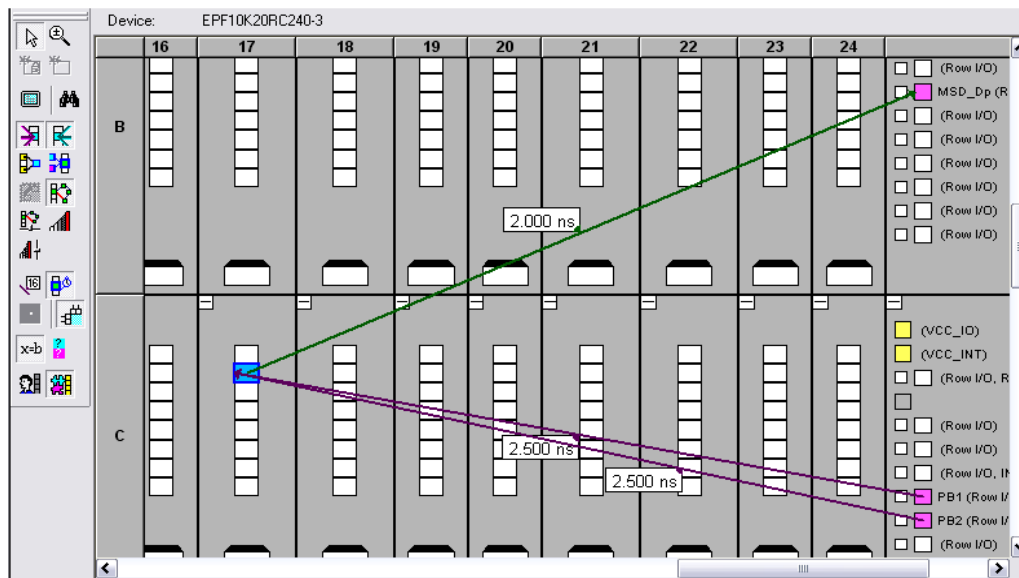
norhradlo.pin – súbor, kde sú definované priradenia pinov

norhradlo.done – čas a dátum vytvorenia projektu

norhradlo.pof – súbor POF obsahujúci údaje na programovanie

norhradlo.sof – súbor SOF obsahujúci údaje na programovanie SRAM

Po kompilácii môžeme otvoriť **Floorplan editor** (obr.15), ktorý slúži na zobrazenie plánu priestorového usporiadania obvodu (čipu) - usporiadanie buniek, makrobuniek, LAB. V okne môžeme sledovať všetky umiestnenia a prepojenia v obvode, čas jednotlivých prepojení, kritické prepojenia, vstupné a výstupné vetvenia.

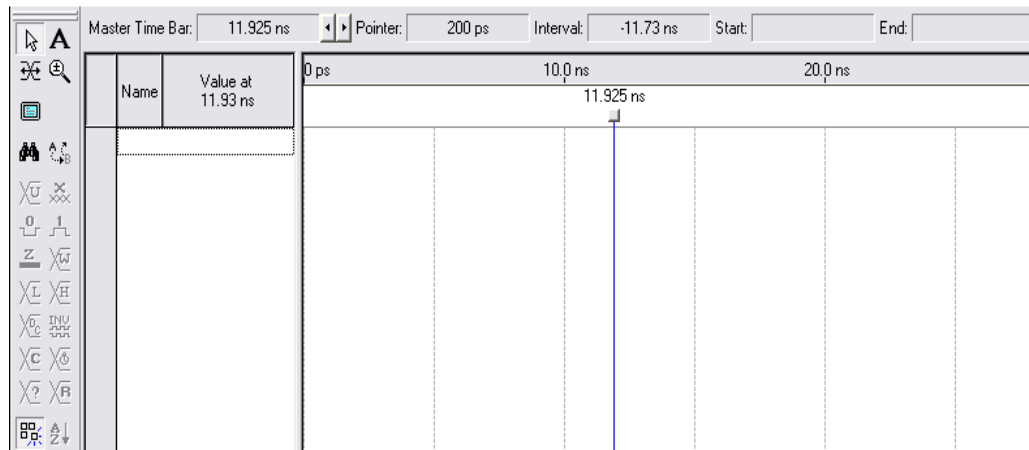


Obr.15: Floorplan editor

2.3.5 SIMULÁCIA PROJEKTU

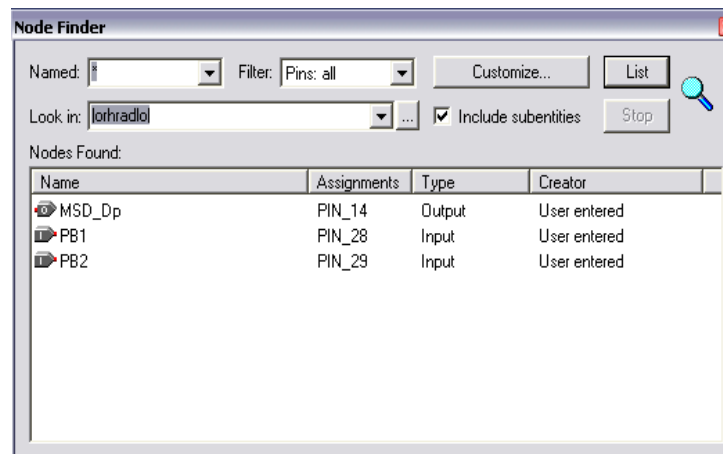
Po úspešnej kompilácii môžeme projekt simulovať. Simuláciu vykonáme nasledovným postupom:

- Vytvoríme vektorový súbor priebehu signálov – **Vector Waveform file (.vwf)**, voľbou **New** z **File menu** (obr.16).



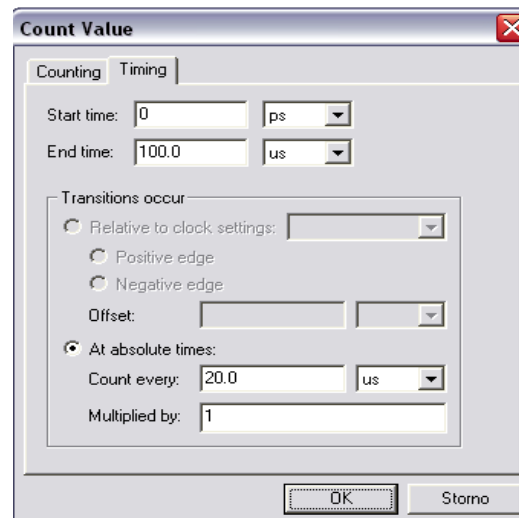
Obr.16: Okno vektorového súboru

- Pomocou voľby **Save As** z **File** menu uložíme tento vektorový súbor, v tomto prípade s názvom *norhradlo.vwf*
- Pomocou **Node Finder** (obr.17) vložíme do tohto súboru všetky vstupy a výstupy (stimuly), ktoré chceme simulovať. **Node Finder** otvoríme nasledujúcim postupom: **View** → **Utility Windows** → **Node Finder**. Onačíme požadované piny (Shift + ľavé tlačidlo myši), skopírujeme ich (Ctrl+C) a vložíme do vektorového súboru (Ctrl+V). Druhou možnosťou je chytiť myšou požadovaný pin a jednoducho ho presunúť do vektorového súboru (.vwf). V našom prípade presunieme piny: PB1, PB2 a MSD_dp.



Obr.17: Okno Node finder

- Nastavíme koncový čas simulácie. V položke **Edit→End Time**, nastavíme hodnotu **time** na 100 μ s.
- Klikneme pravým tlačidlom myši na pin PB1 V tabuľke ktorá sa objaví vyberieme **Value→Count Value**. Zobrazí sa okno (obr.18), kde zdefinujeme v záložke **Timing→End Time**: 100 μ s a **Count Event**: 20 μ s. Pre pin PB2 postup opakujeme a zdefinujeme: **End Time**: 100 μ s a **Count Event**: 10 μ s. (Môžeme zdefinovať aj iné hodnoty).

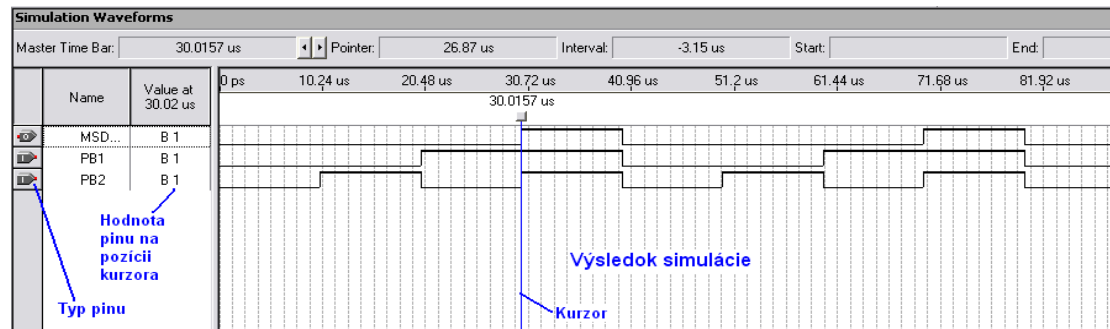


Obr.18: Okno definovania hodnôt časov pre pin PB1

- Uložíme nastavenia voľbou **Save** z **File menu**, alebo prostredníctvom ikony na panely nástrojov. Potom spustíme simuláciu voľbou **Start Simulation** z **Processing menu**, alebo prostredníctvom ikony na panely nástrojov.

Po spustení simulácie sa automaticky zobrazia nasledujúce okná:

- Status
- Simulation Report (má pravé a ľavé okno), (obr.19)
- Message Window



Obr.19: Pravé okno Simulator Report

Z výsledkov simulácie môžeme vidieť, kedy bude LED rozsvietená a kedy zhasnutá. Keď sú tlačidlá PB1 a PB2 keď sú stlačené majú na svojom výstupe logickú úroveň 0. Dióda LED svieti vtedy, keď je na ňu privedená logická úroveň 0. Uvedené skutočnosti musíme uvažovať pri vyhodnocovaní priebehov vo waveform editore.

Charakteristika týchto okien je rovnaká ako to bolo pri kompilácii. Simulácia má tiež svoje moduly:

- Netlist Builder
- Simulator

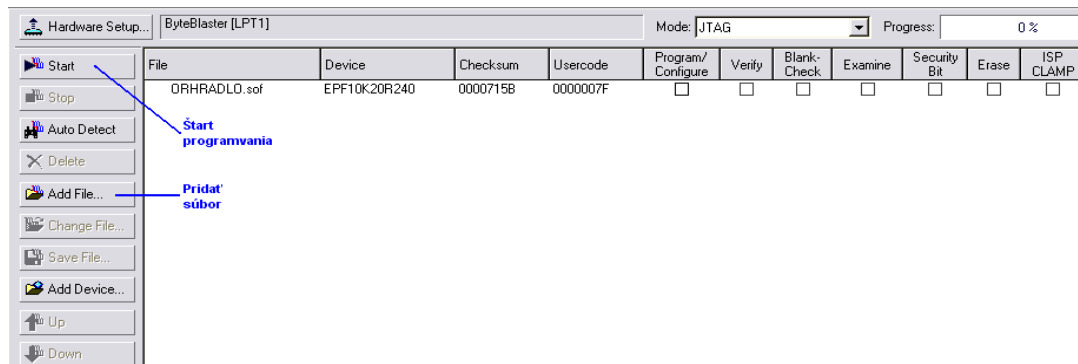
Pri simulácii bol vytvorený nový súbor:

norhradlo.sim.rpt – súbor správ, ktoré generuje simulátor

2.3.6 PROGRAMOVANIE/KONFIGURÁCIA

Po úspešnej kompilácii môžeme programovať obvod voľbou **Tools→Programmer**. Po spustení sa otvorí okno (obr.20) programátora, prostredníctvom ktorého vytvoríme súbor:

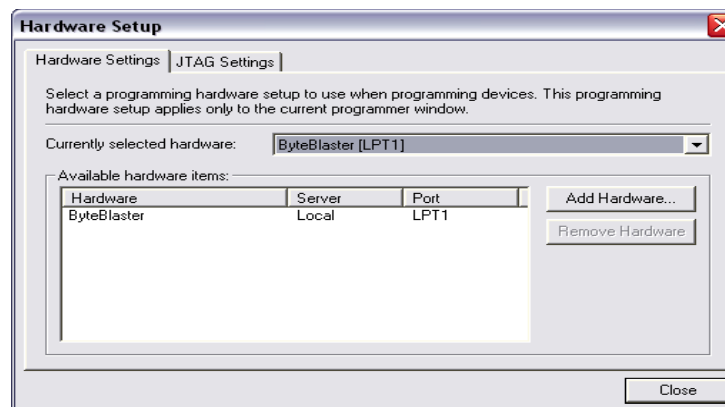
.cdf – **chain description file** – súbor obsahujúci názov a nastavenia obvodov použitých v návrhu



Obr.20: Okno programovania/konfigurácie

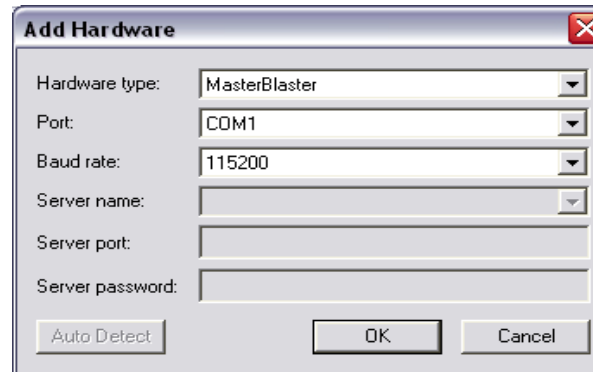
V tomto okne, pred samotným programovaním, musíme nastaviť položky **Hardware Setup** a **Mode** nasledovným postupom:

- Kliknutím na položku **Hardware Setup** sa objaví okno hardvérovej konfigurácie (obr.21)



Obr.21: Okno hardvérovej konfigurácie

- V zobrazenom okne, klikneme na položku **Add Hardware**
- V riadku **Hardware Type** vyberieme **ByteBlasterMV or ByteBlaster II** (podľa typu použitého ByteBlastra) a v riadku **Port: LPT1**



- Nastavenia potvrdíme tlačidlom **OK**.
- Nastavíme JTAG mód

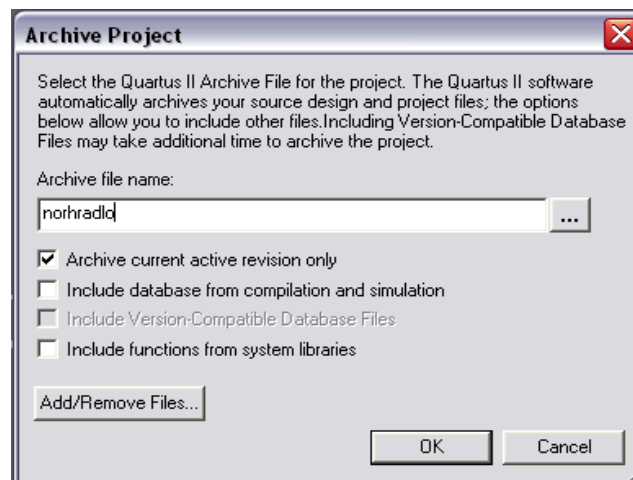
Obvod naprogramujeme pomocou súboru *norhradlo.sof*, ktorý pridáme do programátora pomocou ikony **Add File** (Pridaj súbor). Po pridaní súboru programácie zaškrtneme políčko **Program/Configure** a spustíme programovanie ikonou **Start Programmer** (Štart Programovania). Po spustení sa automaticky otvorí okno hlásení, v ktorom sa objavia správy o programovaní.

2.3.7 ARCHIVÁCIA

V procese tvorby, kompilácie, simulácie a programovania projektu do súčiastky, návrhový prostriedok Quartus II vytvára množstvo súborov. Pri prenose projektu to môže spôsobovať problémy. Preto Quartus II obsahuje takzvanú archiváciu projektu, ktorá vytvorí jeden súbor s informáciami potrebnými k obnoveniu projektu po prenose. Po obnovení projektu a vykonaní kompilácie a simulácie dostaneme pôvodné súbory. Súbory obsahujúce informácie o simuláciách nie sú potrebné k prenosu, lebo po obnovení projektu môžeme vykonať simulácie. Potom dostaneme pôvodne súbory. Simulácia je vývojový prostriedok slúžiaci k overeniu funkčnosti návrhu. Nieje potrebný k naprogramovaniu návrhu do obvodu.

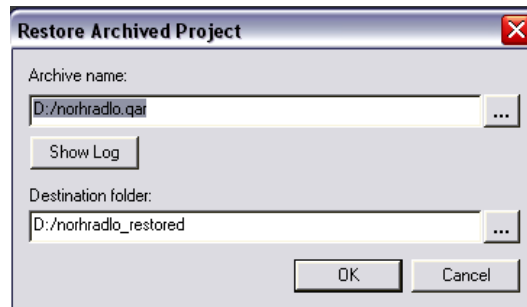
2.3.7.1 POSTUP PRI ARCHIVÁCII PROJEKTU

V položke **Project** vyberieme **Archive Project**. Zobrazí sa okno podľa obr. 22. Do riadku **Active File Name** zadávame názov archivačného súboru v tomto prípade *norhradlo*, poprípade aj cestu, kde má byť súbor uložený. Potom si môžeme vybrať z troch možností, podľa toho, ktoré súbory potrebujeme pridať do archivačného súboru. Prvou možnosťou je archivácia súborov nutných k obnove projektu. Tieto súbory nám postačia v prípade, ak môžeme po obnove archivačného súboru vykonať plnú kompiláciu a simuláciu. Druhou možnosťou je vloženie súborov kompilácie a simulácie a treťou je vloženie funkcií zo systému knižníc. Archivačný súbor má príponu *.gar*. Po stlačení OK bude vygenerovaný archivačný súbor *norhradlo.gar*.



Obr. 22: Okno archivácie projektu

Po prenose archivačného súboru, napr. na iný počítač, vyberieme v návrhovom prostriedku Quartus II možnosť z **File menu** → **Open Project**. Vyberieme archivačný súbor *norhradlo.gar*. Po jeho otvorení sa nám zobrazí dialógové okno (obr.23), kde máme názov archivačného súboru a cieľ, kam sa má projekt rozbaľiť. V riadku **Destination Folders** nastavíme cestu k adresáru, kde sa má projekt umiestniť. Klikneme na **OK**.



Obr.23: Okno obnovy projektu

2.4 POUŽITÍM HRADIEL OR, AND, NAND – PŘÍKLAD Č.2

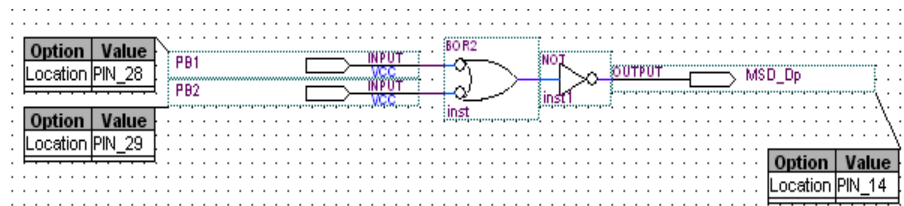
Vo vývojovom prostredí Quartus II realizujte projekt na rozsvietenie desatinnej bodky MSD (MSD_dp) sedem segmentovej LED stlačením jedného z tlačidiel (PB1, PB2) na vývojovej doske UP1 CPLD. Projekt realizujte použitím hradiel OR, AND, NAND, v grafickom editore softvéru Quartus II.

2.5 RIEŠENIE

Postup riešenia bude podobný ako v prípade hradla NOR (kap. 2.2). Taktiež platí tá istá tabuľka pinov ako v prípade hradla NOR (tab.1). Opäť otvoríme nový projekt, podľa postupu uvedeného pre hradlo NOR. Rozdiel je v tom, že pri výbere prvku si nevyberieme hradlo NOR, ale hradlo OR, AND alebo NAND, podľa zadania príkladu. Priradenie pinov, kompiláciu, simuláciu a programovanie vykonáme podľa postupu popísaného v predchádzajúcich kapitolách 2.3.3 – 2.3.6.

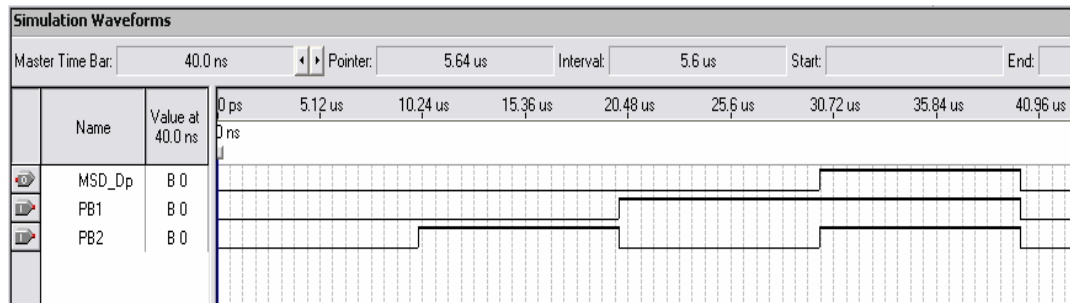
OR

Výsledná schéma s použitím hradla OR je na obr.24. V tomto prípade sme museli dať na výstup ešte invertor, ináč by sme dostali presne opačný efekt. LED by bola stále rozsvietená a zhasla by pri stlačení tlačidla PB1 alebo PB2.



Obr.24: Výsledné zapojenie s použitím hradla OR

Po vykonaní simulácie dostaneme priebehy podľa obr.25.

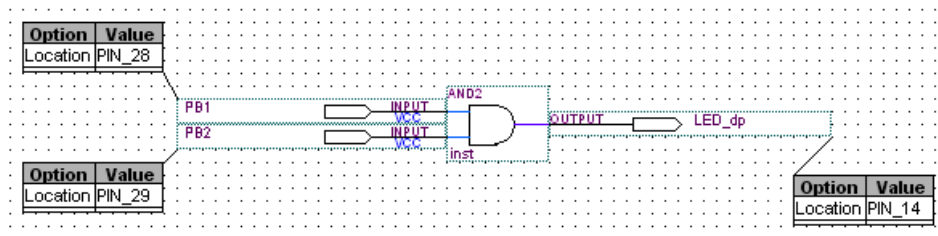


Obr.25: Výsledok simulácie

Naprogramovanie vykonáme podľa postupu uvedeného pri hradle NOR.
(kap. 2.3.6)

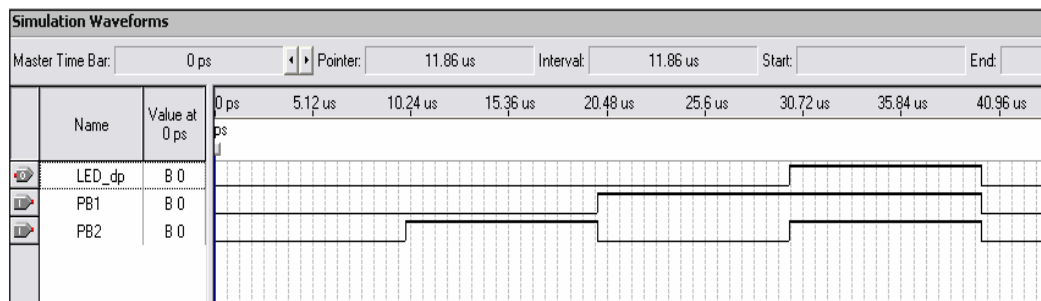
AND

Hradlo AND realizuje logický súčin. Postup návrhu celého projektu je podobný s návrhom projektu pomocou hradla NOR. Výsledná schéma je tvorená dvojjstupovým hradlom AND, dvoma vstupmi (PB1, PB2) a jedným výstupom (LED) (obr. 26).



Obr.26: Zapojenie pomocou hradla AND

Simuláciou tohto zapojenia dostaneme priebehy podľa obr. 27.

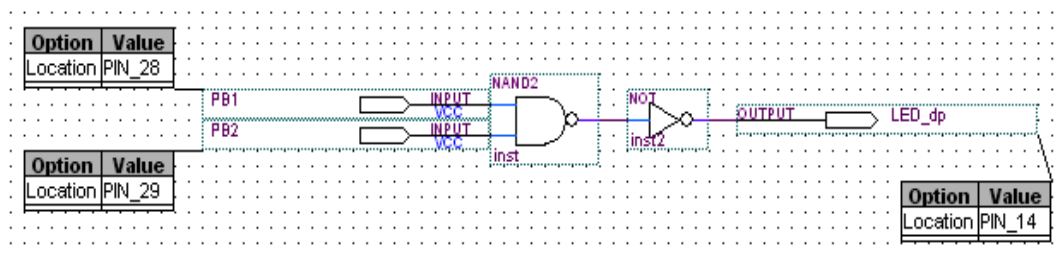


Obr.27: Výsledok simulácie

Programovanie projektu do súčiastky vykonáme podľa postupu uvedeného pri hradle NOR. (kap. 2.3.6)

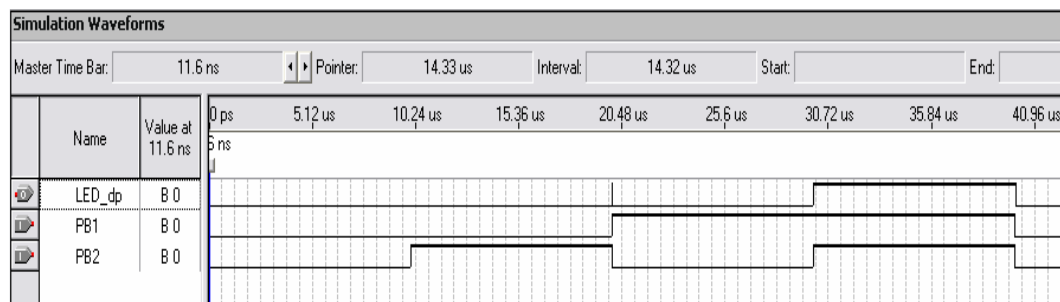
NAND

Hradlo NAND realizuje negovaný logický súčin. Celý návrh je podobný s návrhom ako v prípade hradla NOR. Výsledné zapojenie je tvorené jedným dvojjstupovým hradlom NAND, jedným invertorom, dvoma vstupmi (PB1, PB2) a jedným výstupom (LED) (obr.28).



Obr. 28 Výsledné zapojenie s použitím hradla NAND

Po simulácii dostaneme priebehy podľa obr.29.



Obr. 29 : Výsledok simulácie

Programovanie projektu vykonáme presne podľa postupu uvedeného pri hradle NOR (kap. 2.3.6).