

LABORATORIJSKE VJEŽBE BR. 1

Uvod u Quartus razvojno okruženje.

Intel (Altera) Quartus razvojno okruženje.

Quartus Prime je razvojno okruženje za Intel programibilne čipove i razvojne ploče. Prije nego što je Alteru kupio Intel (2015), ovaj programski paket se zvao Quartus II.

Radi se o platformi koja sadrži sve potrebne alate za svaku fazu dizajna logičkog kola. Postoje više izdanja sa različitim performansama. Quartus Prime Lite je besplatan, ali je zbog svojih ograničenih mogućnosti predviđen za CPLD i jednostavnije FPGA.

Ove pripreme će ukratko prikazati korištenje Quartus Prime dizajn softvera za kreiranje jednostavnog FPGA projekta, kompajliranje projekta sa izlaznom i konačno downloadom fajla u interni RAM ciljane FPGA ploče. Dodatno će biti objašnjeni koraci programiranja eksternog uređaja preko JTAG sučelja.

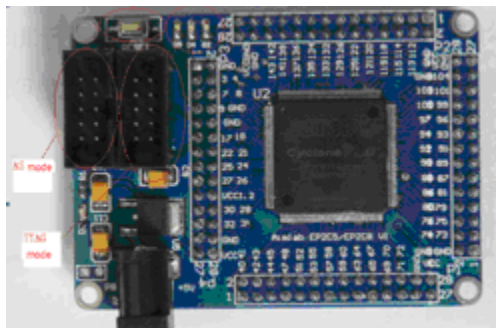
Prije početka rada potvrditi sljedeće:

1. Instalirati Quartus Prime Lite softver koji podržava željeni FPGA čip
- download softvera sa Intel Web
Direktni link za download (5.9 GB windows verzija):
<https://www.intel.com/content/www/us/en/software-kit/661019/intel-quartus-prime-lite-edition-design-software-version-20-1-for-windows.html>
2. Instalirati USB Blaster driver - Ukoliko računar ne prepozna USB Blaster nakon što ga priključite na računar (provjeriti u Device Manageru), morate ga reinstalirati pomoću drivera koji se nalazi u **intel\<verzija> \quartus\drivers** folderu.

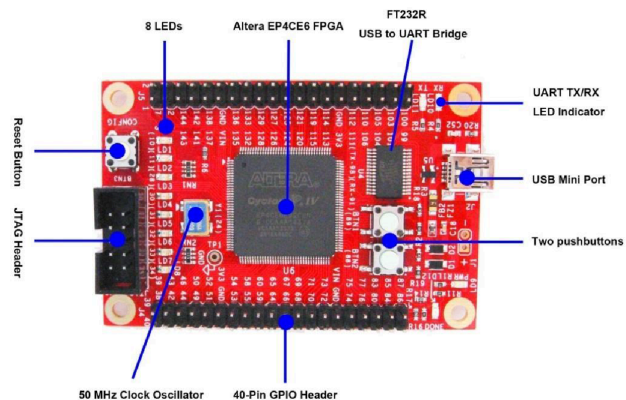
1.1. Dostupne ploče u laboratoriji

U laboratoriji je dostupno nekoliko razvojnih ploča sa različitim karakteristikama. Ovdje su navedene osnovne koje nam trebaju u vježbi, koje će kasnije biti proširene u narednim vježbama:

Naziv	Čip	LED diode	Tipke	Kristal 50MHz
EPM240 Mini	EPM240T100C5	nema	nema	64
EP4 Mini	EP4CE6E22C8N	10,11,28,30, 31,32,33,34	88,89	24
EP4 Starter	EP4CE6E22C8N	7,3,2,1,144, 143,142,141	23,24,25	91
EP2 Mini	EP2C5T144C8N	3,5,7	144	17
De0 Nano Soc	5CSEMA4U23C 6N	W15, AA24, V16, V15, AF26, AE26, Y16,AA23	Switchevi: H5, H6, L9, L10 Pushbuttons AH17,AH16	V11, Y13, E11



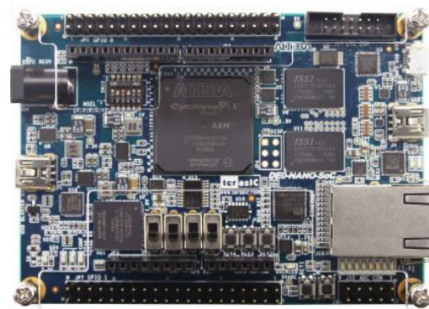
EP2 PLOČA



EP4 MINI PLOČA



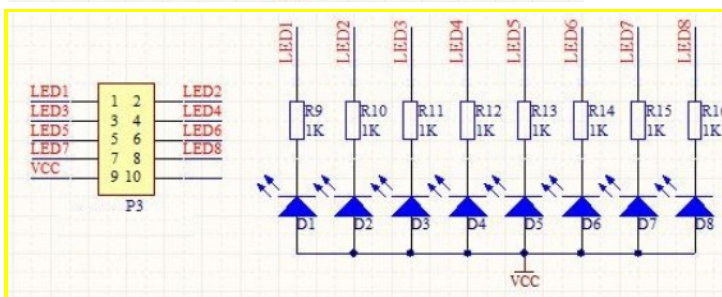
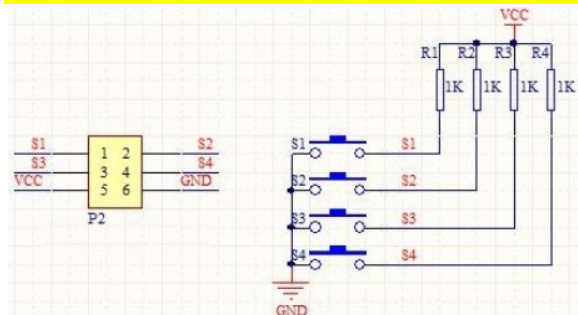
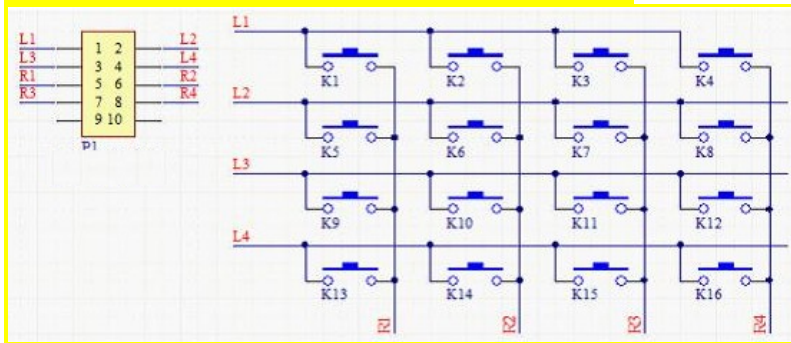
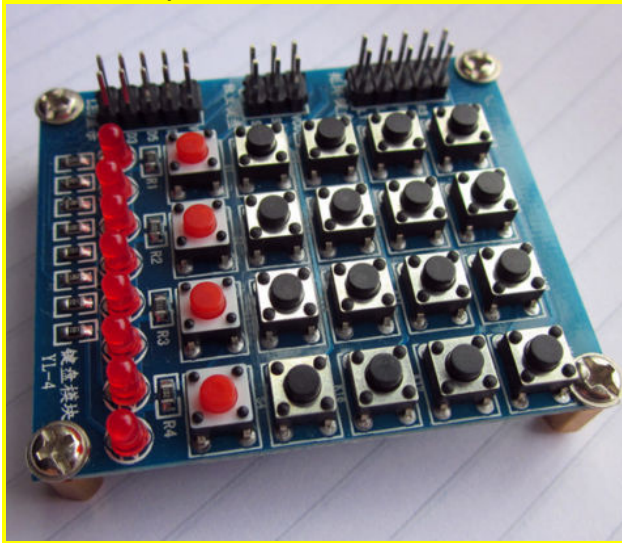
EPM240 PLOČA



De0 NANO SoC

Dodatno, osim razvojnih ploča će se koristiti ploča sa tipkama i LED diodama, 7 segmentni displej, RGB diode i switchevi.

Ploča sa tipkama:



1.2 Kreiranje novog Quartus Projekta

1. **File > New Project Wizard...**
2. Odabrati direktorij za projekt i naziv projekta.
3. Identično ime će se automatski dodijeliti top-level design entitetu.
4. Next (ne koristimo dodatne alate za simulaciju)
5. S obzirom da u prvom projektu ne dodajemo dodatne fajlove, kliknuti Next.
6. Odabrati familiju **Max II, Cyclone II ili IV E ili V**, u zavisnosti od ploče koja Vam je dodijeljena i odabrati željeni čip (prema tabeli u poglavlju 2)
7. Postavke u **Show in 'Available device' list** će smanjiti listu ali nisu neophodne.
8. Klikati **Next** do **Summary** stranice, nakon toga **Finish**

1.3 Kreiranje Verilog fajla

```
module blink (clk, led);  
input clk;  
output reg led;  
reg [25:0] counter;  
  
always@ (posedge clk)  
begin  
counter <= counter+1;  
if (counter>=26'b01111111111111111111111111111111)  
led<=1;  
else led<=0;  
end
```

1.4. Kompajliranje projekta

Nakon što snimimo verilog fajl, možemo kompajlirati projekt.

Važno je napomenuti da je kompajleru potrebno naglasiti koji je „Top-level entity“ - glavni fajl od kojeg započinje kompajliranje (ekvivalent main funkciji u c-u).

- Ukoliko se verilog fajl zove isto kao i naziv modula unutar tog fajla, kompajler će pretpostaviti da je to glavni fajl

- u protivnom je potrebno odabrati TLD entity u prozoru Assignments>Settings u General tabu. To će u ovom slučaju biti blink.

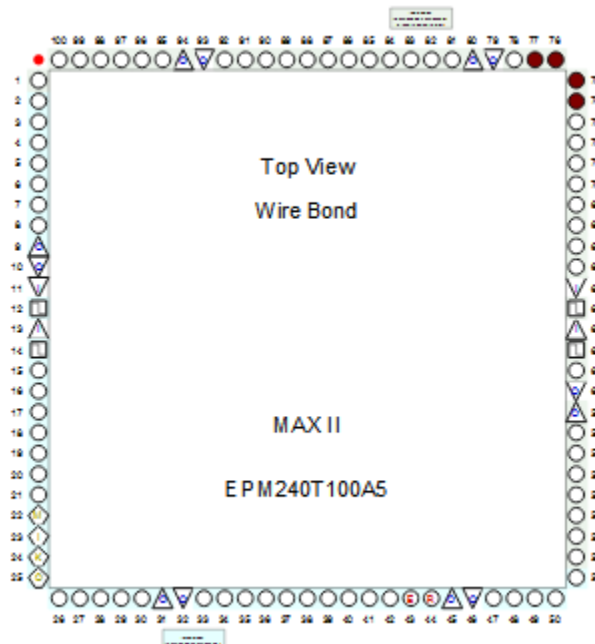
Nakon izvršenog kompajliranja, ukoliko nema grešaka, možemo pristupiti sljedećem koraku

1.5 Pin assignment alat

Nakon što smo napisali verilog kod, možemo pristupiti downloadu kola na naš programabilni čip. Prvi korak je ispravno odabrati koji pinovi će nam biti ulazi a koji izlazi. Neke od dostupnih ploča imaju preklopke, tipke i LED diode na sebi, koje će nam pomoći u simulaciji, a za druge ćemo dodati pločicu sa tipkama i LED diodama.

Pokrenuti **Assignments > Pin planner** alat. U koloni **Location** postaviti odgovarajuće brojeve pinova (prema željenom rasporedu). Pinove koje odaberete (npr: 74 za LED diodu i 64 za clock – ODABIR VRŠITI PREMA TABELI iz 2.1 i PLOČI KOJA VAM JE DODIJELJENA) povezati sa led diodama. Clock se nalazi interno povezan na ploči na pinu koji je naznačen u tabeli i nije ga potrebno dodatno povezivati.

Smeđom bojom su označeni iskorišteni pinovi:

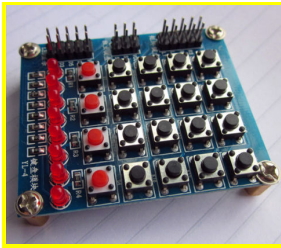


1.6. Povezivanje hardvera

Ukoliko se ne koriste LED diode na ploči, u ovom koraku potrebno je povezati razvojnu ploču sa pločom koja sadrži LED diode:

Spojiti VCC na P3.VCC

Spojiti pin 75 na pin P3.LED1



1.7. Programiranje uređaja

1. Toolbar tipka **Start Compilation** za kompilaciju (ukoliko su vršene promjene u PIN Planneru potrebno je rekompajlirati kod) i generiranje konfiguracijskog fajla (**project_name.sof**).
2. Aktivirati **Programmer** alat na istom toolbaru. Konfiguracijski fajl se automatski dodaje
3. Ako je prikazano **No Hardware**, kliknuti **Hardware Setup...** tipki.
4. Kliknuti **Add Hardware...** tipku i odabrati **Hardware type - Blaster, te Port USB0/USB1**.
5. Kliknuti **OK**, zatim **Close**.
6. **No Hardware** poruka bi trebala se pretvoriti u **Blaster [USBx]**.
7. Postaviti **Mode** na **JTAG**,
8. Selektirati **Program/Configure**
9. Kliknuti **Start**.
10. FPGA je sada konfigurirana. Testirati dizajn. Pritiskati tipke i pratiti izlaz na LED diodama.

Primjer 2:

Modificirati BLINK program tako da ako je pritisnuta tipka, blinka neka druga dioda

```
module blink (clk, led1, led2, tipka);  
input clk, tipka;  
output reg led1, led2;  
reg [25:0] counter;  
reg s;
```

```

always@ (posedge clk)
begin
counter <= counter+1;
if (counter>=26'b01111111111111111111111111111111)
s<=1;
else s<=0;

led1<=0;
led2<=0;
if (s) led1<=s;
else led2<=s;
end

```

Primjer 3:

Modificirati BLINK program tako da ako je pritisnuta tipka, dioda blinka dvostruko brže.

```

module blink (clk, led, tipka);
input clk, tipka;
output reg led;
reg [25:0] counter;

always@ (posedge clk)
begin
counter <= counter+1;
if (tipka)
led<=counter[26];
else
led<=counter[25];
end

```