

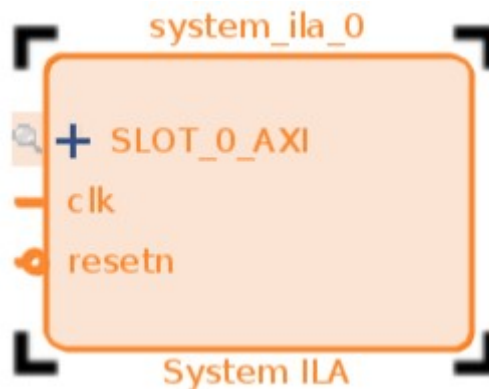
Integrated Logic Analyzer (ILA)

Uvod

U određenim situacijama neophodno je debugovati digitalni sistem nakon programiranja FPGA čipa, jer je u simulaciji potencijalno promakao neki bag. Da bi se ovo omogućilo kompanija Xilinx je obezbedila ILA IP blok. To je dodatni hardver koji može da nadgleda bilo koji port ili signal u digitalnom sistemu i da informacije o tome šta se dešava na njemu vraća preko serijskog interfejsa našem računaru.

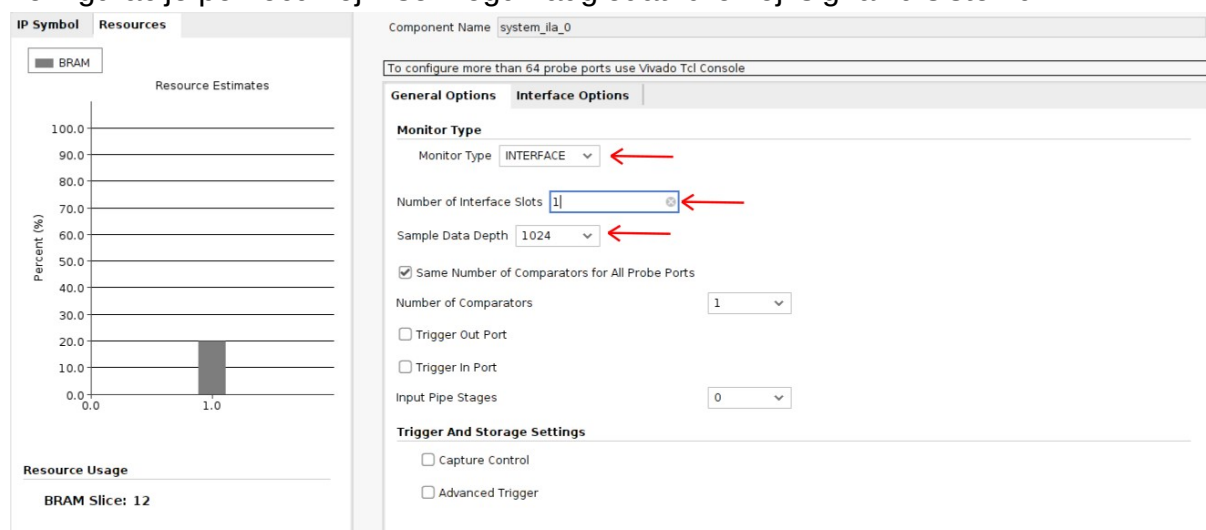
IP integrator

Blok koji je neophodno dodati u IP integratoru zove se **System ILA** i kada se prvi put ubaci ima sledeći izgled:



Slika 1. ILA

Ako se ne napravi nikakva promena u konfiguraciji ovog modula on može da nadgleda samo signale AXI FULL interfejsa. Dvoklikom se otvaraju dodatne konfiguracije pomoću kojih se mogu nadgledati bilo koji signali u sistemu:



Slika 2. ILA config *general options*

Na prethodnoj slici najbitnija opcija je Monitor Type za koju postoji 3 izbora:

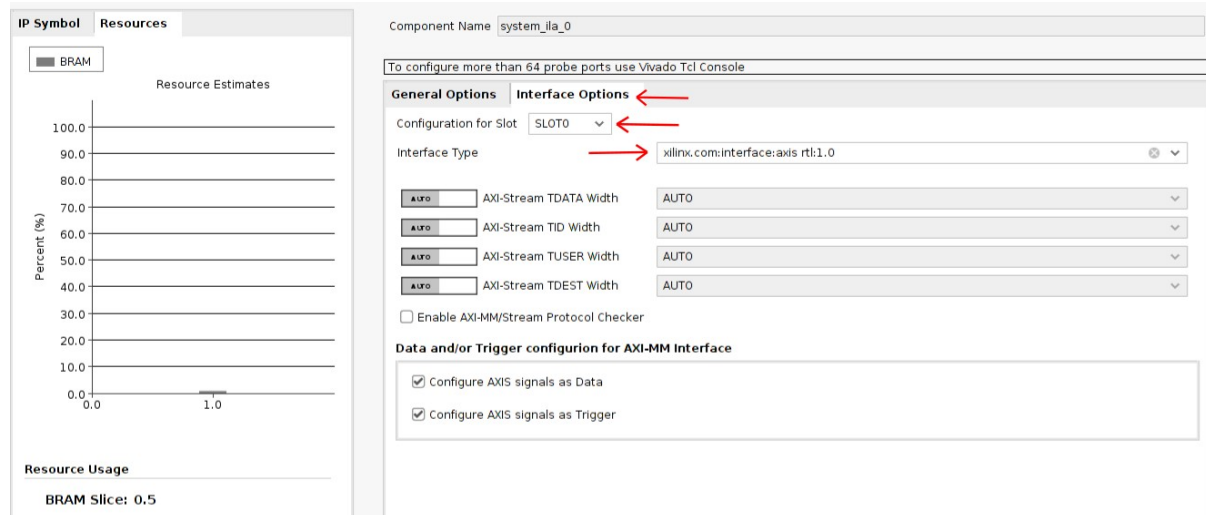
- INTERFACE
- NATIVE
- MIX

INTERFACE ILA

Kada se monitor Type podesi na ovu opciju to znači da hoćemo da nadgledamo standardne interfejske kao što su: AXI FULL, AXI LITE, AXI STREAM, APB, itd. Odnosno ako se odabere ova opcija ILA će nam automatski napraviti portove na sebi za odgovarajući interfejs i njih ćemo koristiti za debugovanje (Na slici 1 je automatski napravljeni portovi za nadgledanje AXI FULL interfejsa).

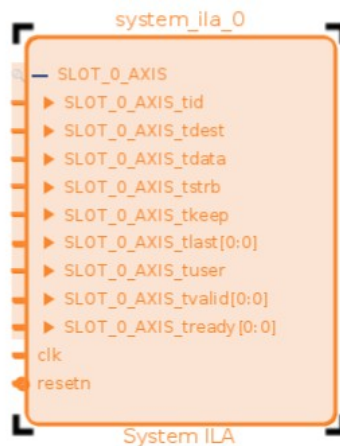
Na slici 2 opcija **Number of Interface Slots** određuje koliko standardnih interfejsa želimo da nadgledamo, dok opcija **Sample Data Depth** određuje koliko rastućih ivica takta od nekog događaja želimo da vidimo, podrazumevano se hvata 1024 odbirka nekog signala.

Nakon što se ove opcije podese odabrati tab *Interface Options*:



Slika 3. ILA interface options

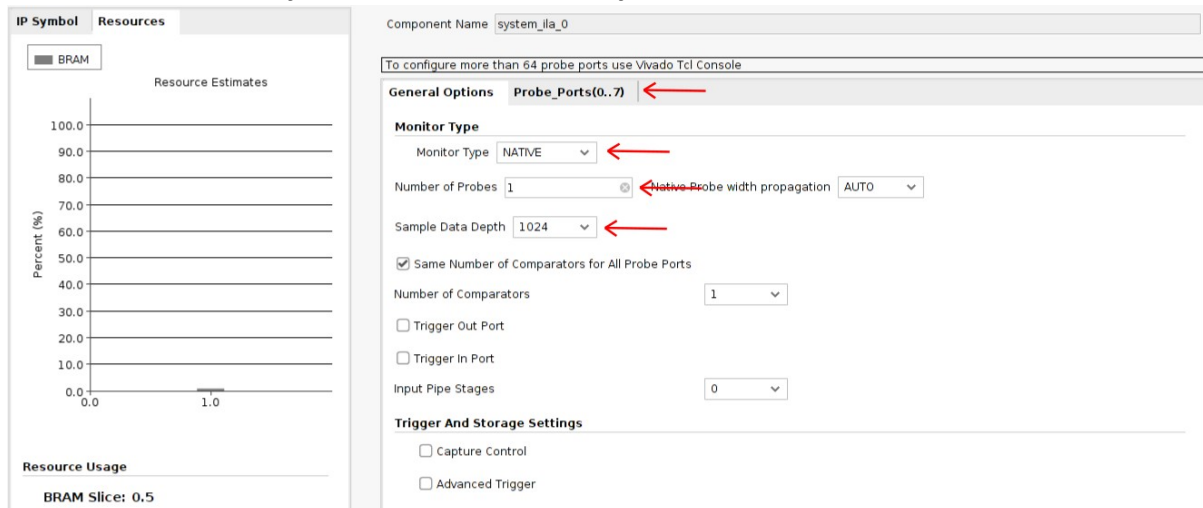
U ovom prozoru može se bira koji interfejs želimo za određeni slot. Odnosno ako je na slici 2 odabrano više slotova u opciji **Number of Interface Slots** možemo da odaberemo više različitih interfejsa koje želimo da nadgledamo. Na slici 3 je za slot 0 odabrano da bude AXI stream interfejs. I ako se nakon ovih podešavanja pritisne OK Interfejs ILA bloka će da izgleda kao na sledećoj slici:



Slika 4. ILA sa AXI stream interfejsom

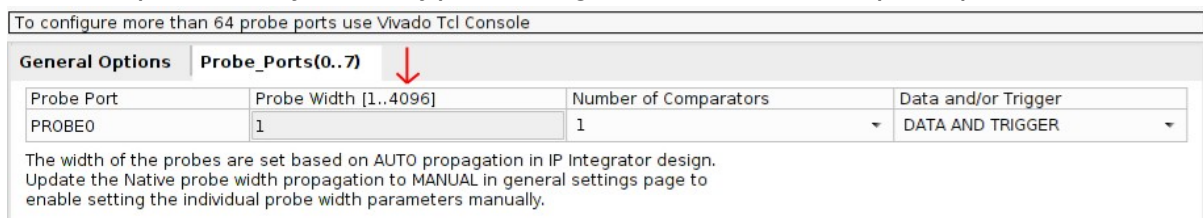
NATIVE ILA

Ukoliko se na slici 2 odabere ova opcija, to znači da treba da se nadgleda neki nestandardni interfejs i slika 2 se malo menja:



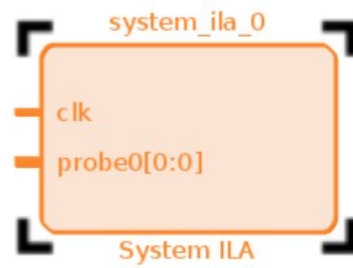
Slika 5. ILA Native konfiguracija

Sada umesto *Number of Slots* opcije imamo **Number of Probes** i umesto *interface options* prozora imamo prozor **Probe_Ports**. Jedina razlika je u tome što je svaki *probe* u stvari port koji će biti povezan na signal koji treba nadgledati. Ako se u opciju *Number of Probes* postavi 1 to znači da posmatramo samo jedan signal. Dodatna podešavanja za ovaj probe mogu se izvršiti u *Probe ports* prozoru:



Slika 6. ILA Native probe config

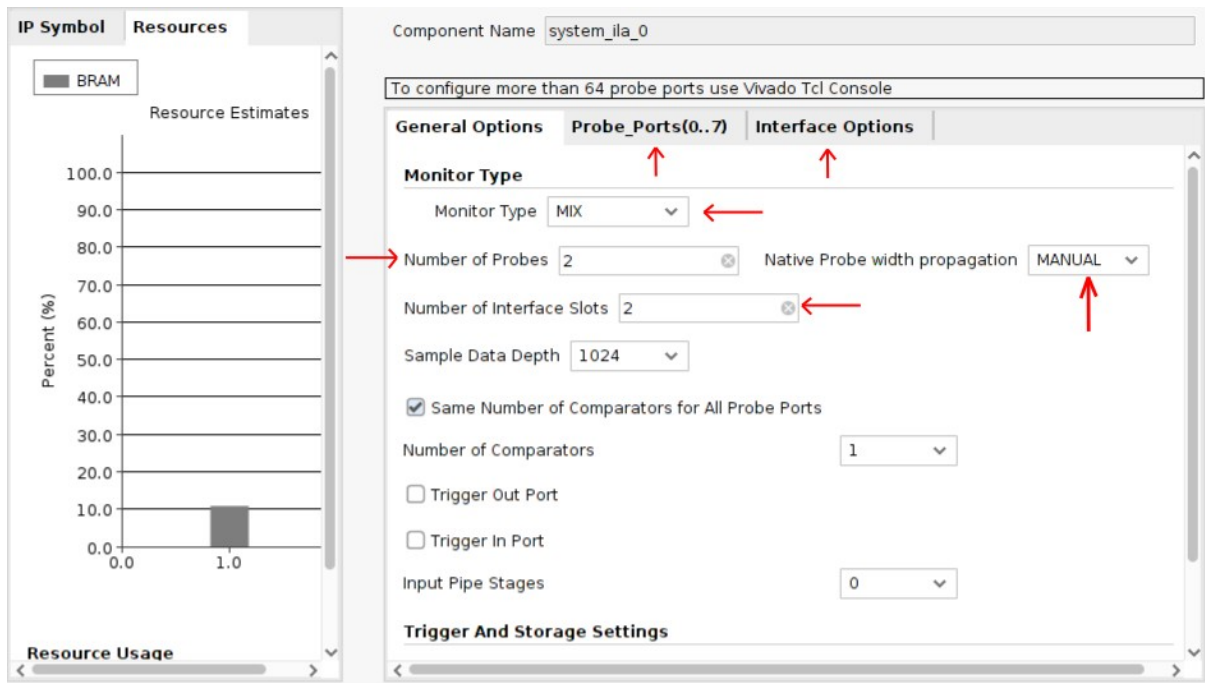
Najbitnija opcija je naznačena crvenom strelicom i ona određuje širinu jednog *probe* porta. Na slici 1 naznačeno je da je širina jedan bit i ako se pritisne ok interfejs ILA bloka će imati sledeći izgled:



Slika 7. ILA sa jednim *probe* portom

MIX ILA

U većini situacija imamo potrebu da posmatramo i standardne i nestandardne interfejske istovremeno tako da ako se odabere opcija MIX u *Monitor Type* dobija se sledeće:

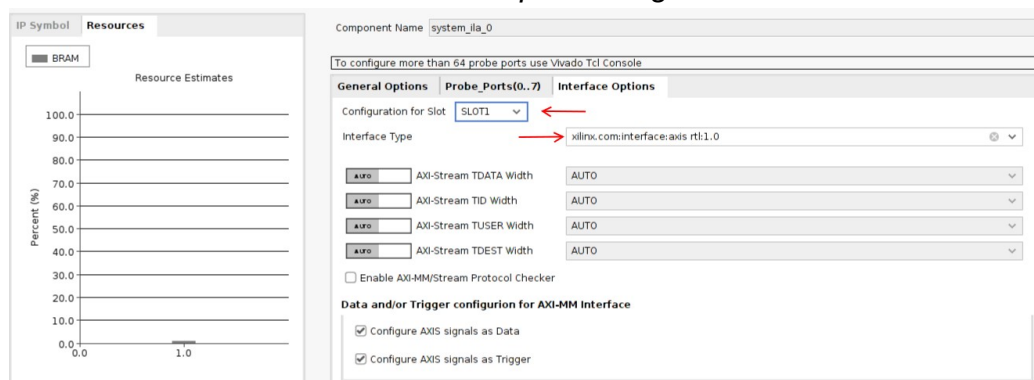


Slika 8. ILA MIX

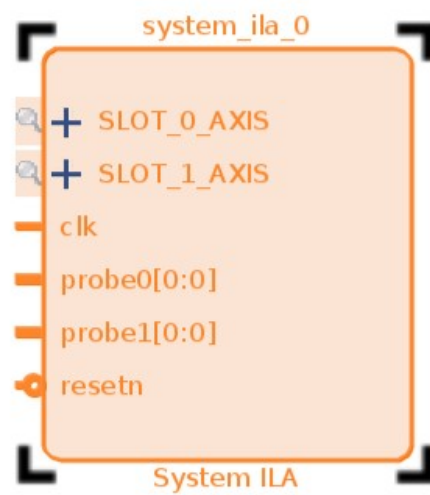
Na slici 8. ILA je konfigurisana tako da ima dva standardna interfejsa (*Number of Slots* = 2) i da nestandardna interfejsa (*Number of Probes* = 2). U pitanju su dva AXI stream interfejsa i dva

To configure more than 64 probe ports use Vivado Tcl Console			
General Options	Probe_Ports(0..7)	Interface Options	
Probe Port	Probe Width [1..4096]	Number of Comparators	Data and/or Trigger
PROBE0	1	1	DATA AND TRIGGER
PROBE1	1	1	DATA AND TRIGGER

Slika 9. ILA MIX port configuration



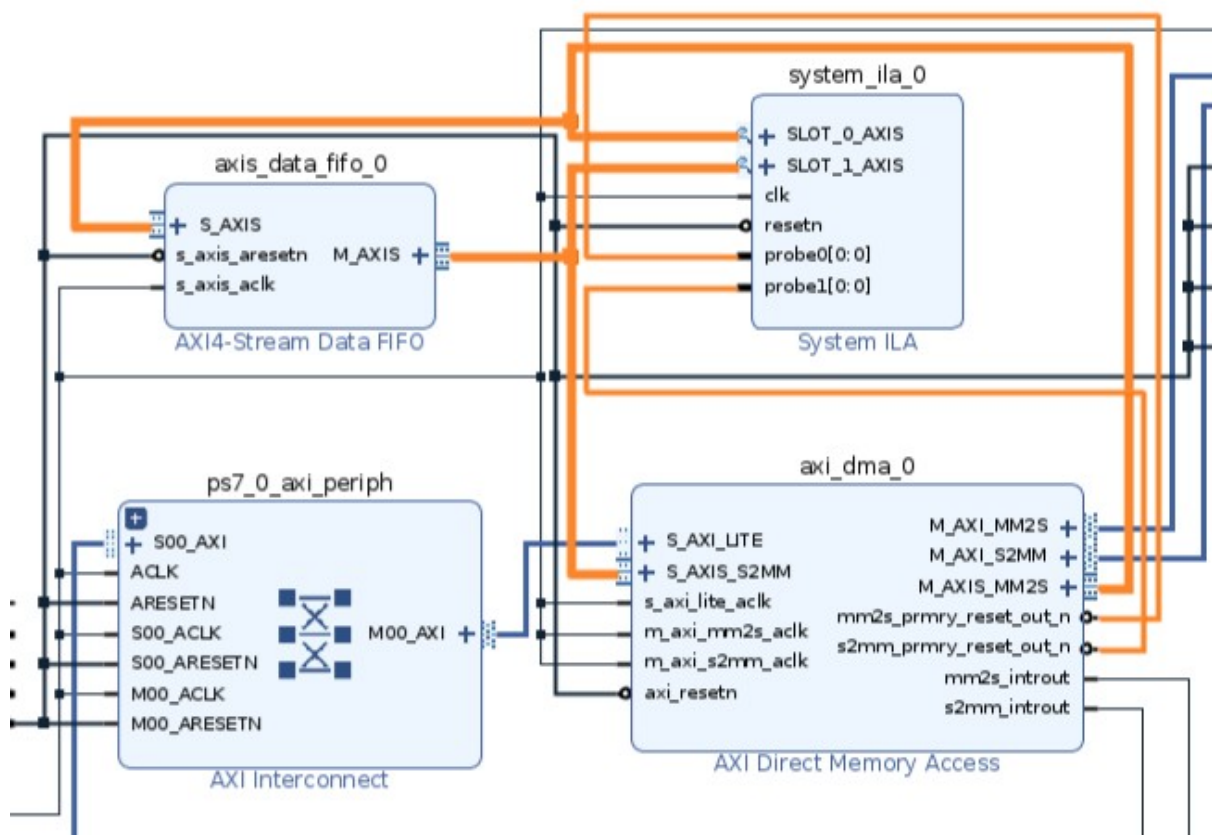
Slika 10. ILA MIX slots configuration



Slika 11. MIX ILA

Primer korišćenja ILA bloka

Način rada ILA bloka će biti prikazan na primeru iz prethodnog materijala. Tu je DMA blok čitao podatke iz DDR memorije, upisivao ih u FIFO blok i te iste podatke ponovo iz FIFO bloka vraćao u DDR. Taj sistem ćemo proširiti sa ILA blokom konfigurisanim kao na slici 11 (dva AXI stream slota i dva jednobitna *probe* porta). Pomoću njega ćemo nadgledati portove koji generišu prekide i transakcije koje se dešavaju na AXI stream interfejsima DMA bloka. Na sledećoj slici je prikazano povezivanje ILA bloka sa DMA blokom:



Slika 12. Povezivanje ILA bloka sa DMA blokom

Neophodno je povezati **SLOT_0_AXIS** i **SLOT_1_AXIS** interfejs ILA bloka sa **S_AXIS_S2MM** i **M_AXIS_MM2S** interfejsima DMA bloka, kao i **probe0** i **probe1** portove ILA bloka sa **mm2s_introut** i **s2mm_introut** portovima DMA bloka. Na ovaj način je omogućeno nadgledanje AXI stream transakcija ali i nadgledanje portova za generisanje prekida. Nakon što je ovo urađeno pokrenuti sintezu i implementaciju.

RUNTIME simulacija

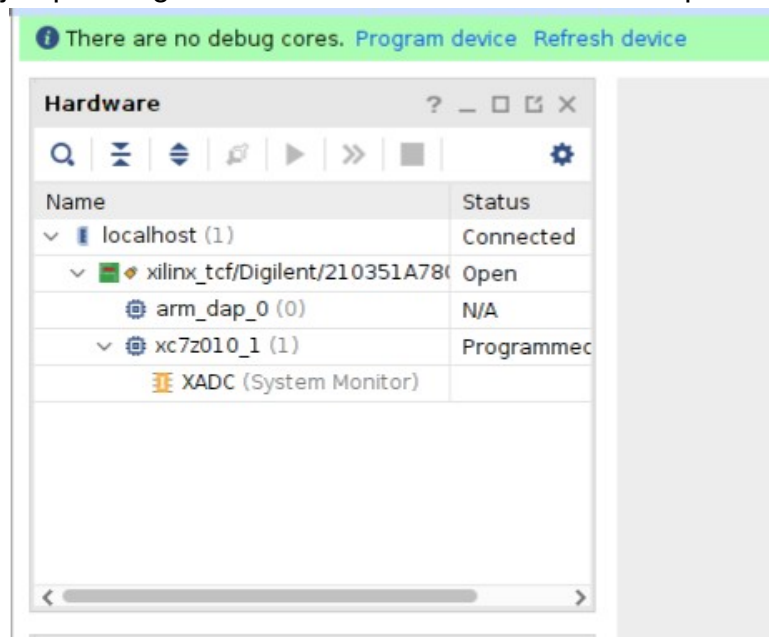
Da bi mogli da se nadgledaju signali za prethodni primer neophodno je koristiti Vivado i Vitis istovremeno. Vivado omogućava posmatranje talasnih dijagrama koje dobija od ILA logike, dok će se Vitis koristiti za pokretanje aplikacije koja će izazvati te talasne dijagrame.

Kada se završi sinteza i implementacija uključiti u Vitis eksportovani hardver i iskoristiti aplikaciju iz prethodnog materijala i izvršiti build za Application project i Platform project.

Sada je u Vivadu neophodno povezati se sa razvojnom platformom:



Slika 13. Povezivanja Vivada sa razvojnom platformom
Odabirom opcije open target i **Auto connect** otvara se sledeći prozor:

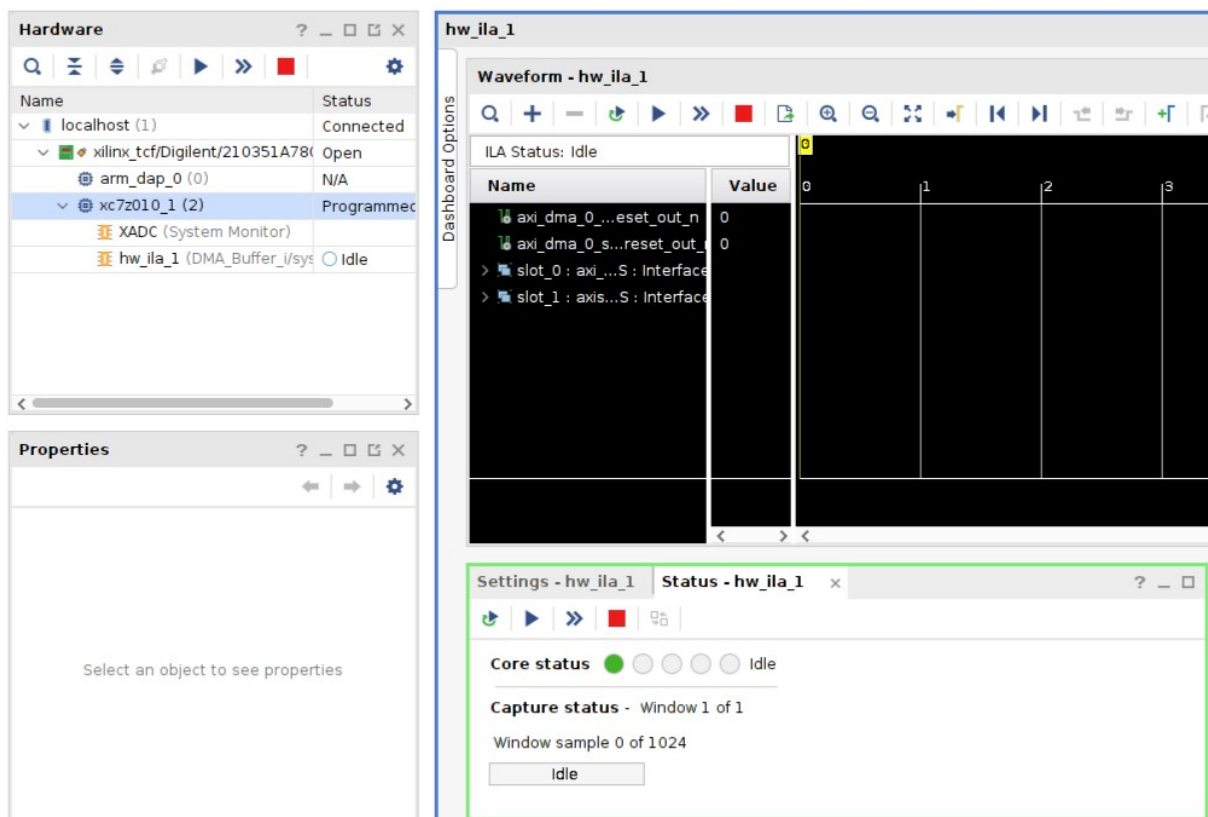


Slika 14. Povezivanje sa razvojnom platformom iz Vivada

U ovom prozoru će biti prepoznat ILA blok i da bi se to omogućilo **neophodno je pokrenuti aplikaciju iz Vitis alata**:

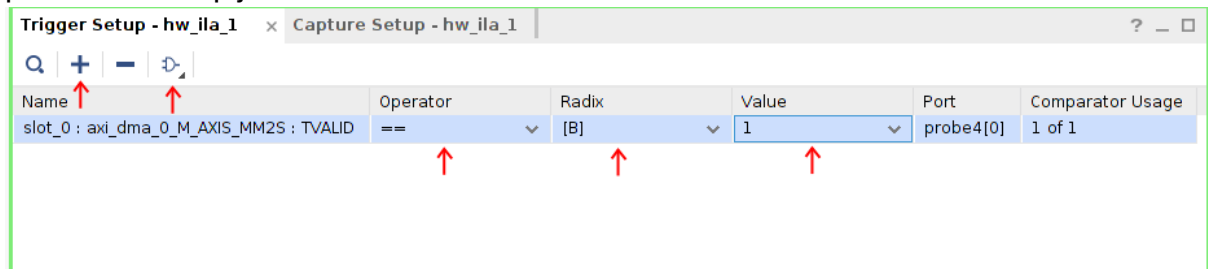
Desni klik na Application Project -> Run As -> Launch Hardware

Kada se pritisne **Refresh Device** sa slike 14 pojaviće se sledeće:



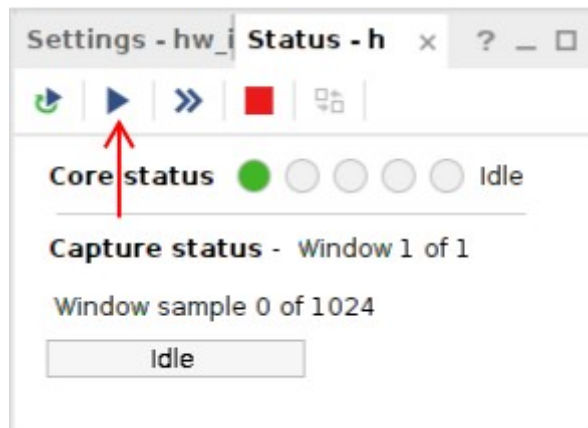
Slika 15. Detekcija ILA bloka u Vivado alatu

Sledeći korak je posmatranje talasnih dijagrama koje dobijamo od ILA bloka. U prozoru *Trigger Setup - hw_ila_1* treba da ubacimo scenario od kog hoćemo da ILA počne da sakuplja odbirke:



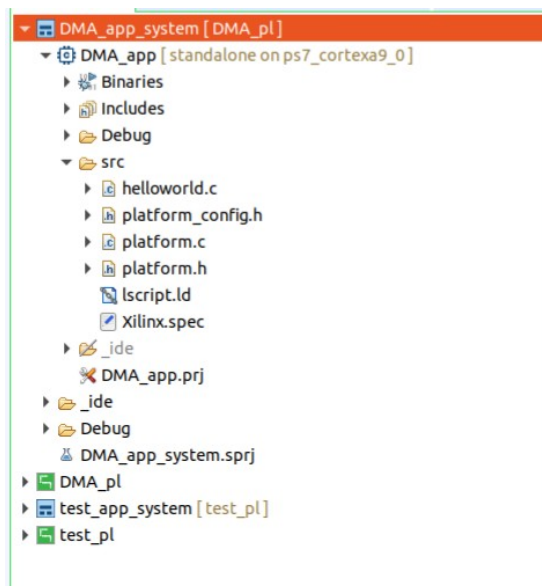
Slika 16. Podešavanje scenarija koji hoćemo da vidimo

Prethodni primer podešava ILA blok tako da uhvati situaciju kada VALID signal M_AXIS_MM2S interfejsa postane logička jedinica, odnosno posmatra se trenutak kada DMA blok pošalje prvi validan podatak ka FIFO bloku. Nakon što se ovo podesi treba pritisnuti sledeće:



Slika 17. "Naoružavanje" ILA bloka

Kada je ILA "naoružana", ona je u pripravnosti i čeka da se dogodi promena TVALID signala sa logičke nule na logičku jedinicu, a to će se desiti tek kada se **opet pokrene aplikacija iz Vitis alata**, ali prvo što je potrebno uraditi jeste podesiti opcije prilikom pokretanja programa iz Vitis-a:

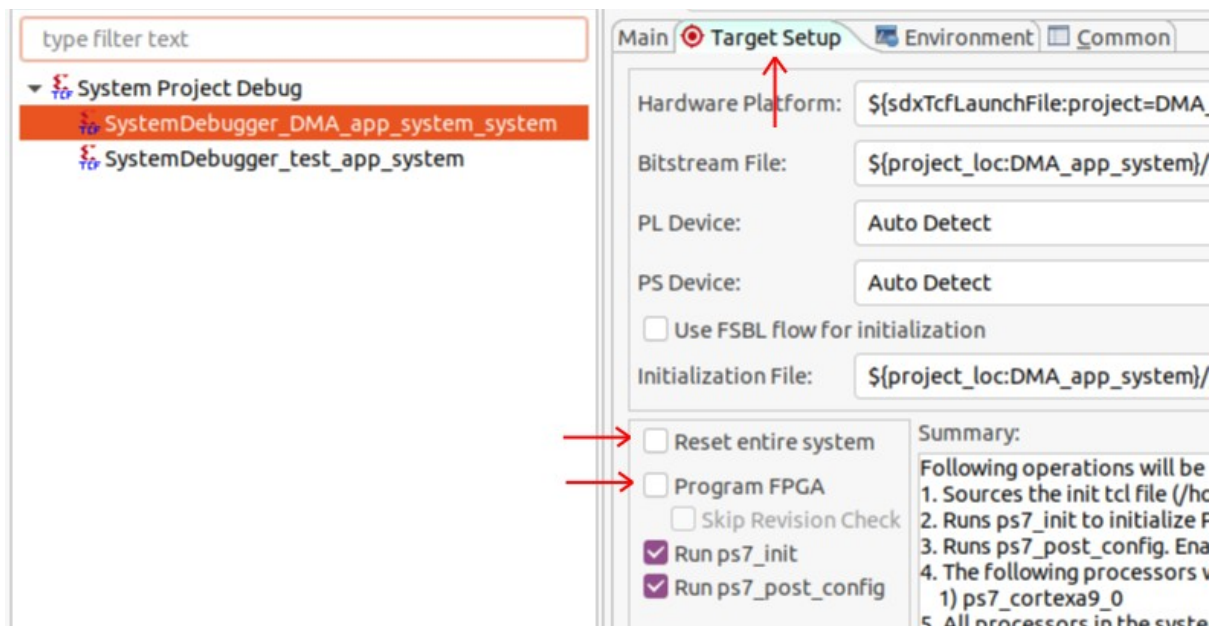


Slika 18. Promena podešavanja za pokretanje aplikacije

Pritisnuti desni klik na application project (DMA_app_system sa prethodne slike) i odabrati:

Run_as->Run_configurations

Isključiti resetovanje i reprogramiranje sistema prilikom pokretanja aplikacije na sledeći način:

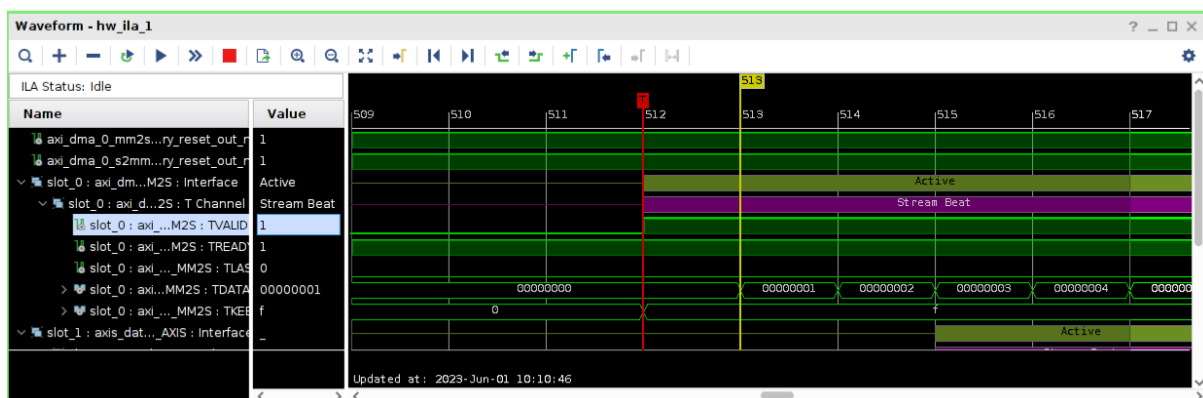


Slika 19. Isključivanje resetovanja sistema

Pokrenuti aplikacije:

Desni klik na Application Project -> Run As -> Launch Hardware

U Vivadu će se pojaviti sledeći talasni signali:



Slika 20. Talasni signale dobijeni od ILA bloka

Sa prethodne slike crvenom strelicom je naznačen trenutak koji smo želeli da uhvatimo (tu je TVALID = 1) i takođe se vide podaci koje iz DDR memorije šaljemo ka FIFO BLOKU (TDATA). ILA blok će sakupiti 1024 odbirka kada se desi odgovarajući scenario podešen na slici 16 (512 odbiraka pre i 512 odbiraka posle željenog trenutka). Prilikom naoružavanja ILA bloka mogu da se smisle u drugačiji scenariji, da se posmatraju drugi signali:

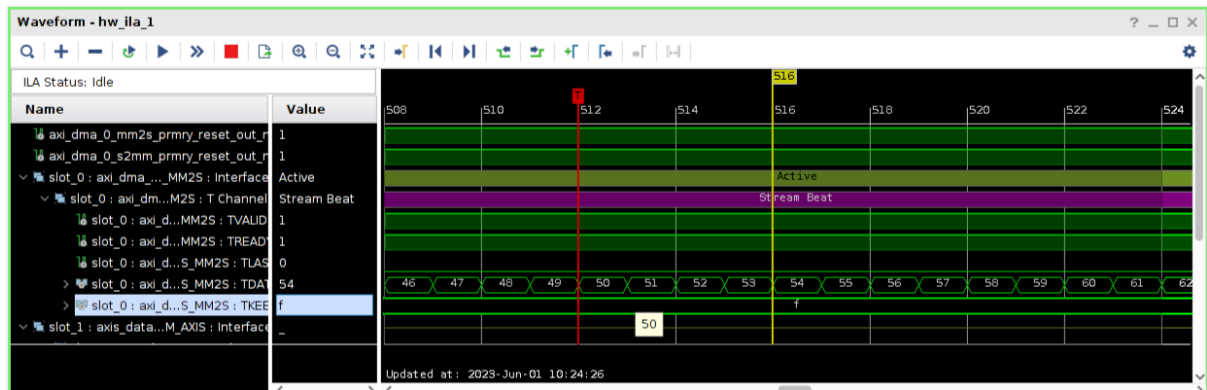
- Da li se desio interapt
- Da li se pojavio određeni podatak.
- Da li se pojavio određeni podatak i TVALID = 1.

- ltd.

Sledeći primer “naoružava” ILA blok tako da čeka trenutak TVALID=1 i TDATA=50:

Trigger Setup - hw_ila_1 x Capture Setup - hw_ila_1						
Name	Operator	Radix	Value	Port	Comparator Usage	
slot_0 : axi_dma_0_M_AXIS_MM2S : TVALID	==	[B]	1	probe4[0]	1 of 1	
slot_0 : axi_dma_0_M_AXIS_MM2S : TDATA	==	[U]	50	probe2[31:(1 of 1	

Slika 21. Primer 2 konfigurisanja ILA bloka



Slika 22. Primer 2 ILA talasni dijagram