



T.C. SANAYİ VE
TEKNOLOJİ BAKANLIĞI

#^{Milli}
TEKNOLOJİ
HAMLESİ



Çip Tasarım Yarışması Mikrodenetleyici Tasarım Kategorisi Silicore

Takım ID: 834669 | Başvuru ID: 4653234



T.C. SANAYİ VE
TEKNOLOJİ BAKANLIĞI



Takım Tanıtımı

Silicore: 4 üye

Bekir Yufka — Takım Kaptanı

Hacettepe Üniv. Elektrik-Elektronik Müh. 4. Sınıf

Mimari tasarım • YZ hızlandırıcı • FPGA implementasyonu

Eyüp Ertan Sağdıç

Hacettepe Üniv. Elektrik-Elektronik Müh. 2. Sınıf

Çevre birimleri • AXI tasarımı • Boot kodu ve firmware • RTL optimizasyonu • Çip tasarım akışı

Arda Tecimen

Hacettepe Üniv. Elektrik-Elektronik Müh. 2. Sınıf

Spike ISS ile çekirdek testleri • Çevre birimi protokol doğrulaması • SoC sistem testleri • Raporlama • AXI tasarımı doğrulama • IVC ve entegrasyonu

Deniz Zafer

Gazi Üniv. Bilgisayar Müh. 3. Sınıf

YZ hızlandırıcı testleri • UVM doğrulama altyapısı



İçindekiler

1 Sistem Mimarisi

2 Tasarım Detayları, Yazılım, Çekirdek ve Bus

3 Tasarım Detayları, Çevre Birimleri ve Boot

4 JTAG ve YZ Hızlandırıcı

5 FPGA Prototipleme ve Demo

6 Doğrulama

7 Çip Tasarım Akışı

8 Kaynakça



Sistem Tanımı ve Minimum İsterlerin Karşılanması

32 bit

CV32E40P · RV32IMC + Zicsr

13 slave

tek AXI4-Lite omurga

47 kB

IMEM + DMEM + YZ + ROM'lar

36 MHz

mcu_clk

45 MHz

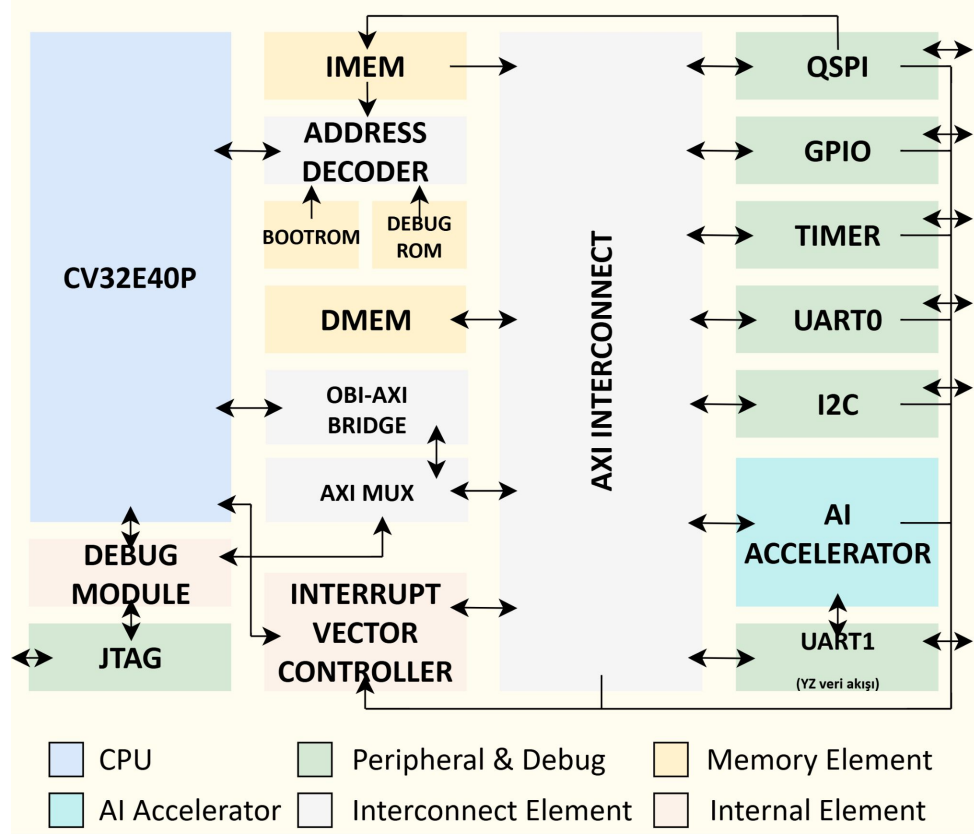
ai_clk

Minimum ve Opsiyonel İsterler — Gerçekleme Durumu

- | | |
|-----------------------------------|---------------------------|
| ✓ CV32E40P çekirdeği | ✓ YZ hızlandırıcı |
| ✓ AXI4-Lite omurga | ✓ YZ iş akışı |
| ✓ GPIO | ✓ JTAG + Debug Module |
| ✓ Timer | ✓ Çekirdek doğrulaması |
| ✓ UART × 2 | ✓ AXI protokol kontrolü |
| ✓ I2C Master | ✓ Sistem seviyesi testler |
| ✓ QSPI Master | ✓ FPGA akışı çıktıları |
| ✓ IMEM 8 kB · DMEM 8 kB · BootROM | ✓ Fiziksel akış → GDSII |
| ✓ QSPI flash'tan boot | |



Sistem Mimarisi



Nihai tasarımın blok şeması

- **Merkezi omurga:** Tüm bellek ve çevre birimleri AXI4-Lite interconnect üzerinden çekirdeğe bağlanır.
- **Komut yolu ayrımı:** Address Decoder Modülü, komut çekme isteklerini adrese göre BootROM, DebugROM veya IMEM'e yönlendirir.
- **IVC:** Tüm çevre birimi olaylarını tek vektörlü kesme hattında toplar, çekirdeğe kaynak kimliğiyle iletir. Aynı zamanda şartnamede istenen çıkarım bitince çekirdek kesme ile bilgilendirilir" isterini karşılamamızı sağlar.
- **Debug:** JTAG + Debug Module, çekirdeği durdurma, adım adım ilerletme ve register erişimi sağlar.
- **Ayrık YZ belleği:** AI Memory sayesinde hızlandırıcı, ana veri yolunu meşgul etmeden çalışır.



Sistem Bellek Haritası

ID	Peripheral	Base Address	Size
1	Boot ROM	0x0000_0000	512 B
2	Debug ROM	0x0000_0200	512 B
3	IMEM	0x0000_1000	8 KB
4	DMEM	0x2000_0000	8 KB
5	Reserved (DMEM Exp)	0x2000_2000	56 KB
6	IVC	0x2002_0000	256 B
7	AI Config	0x3000_0000	256 B
8	GPIO	0x4000_0000	256 B

ID	Peripheral	Base Address	Size
9	TIMER	0x4000_0100	256 B
10	UART0	0x4000_0200	256 B
11	UART1 (Stream)	0x4000_0300	256 B
12	I2C Master	0x4000_0400	256 B
13	QSPI Master	0x4000_0500	256 B
14	Debug / JTAG	0x4000_0600	256 B
15	Test Area (Scratch)	0x5000_0000	1 MB



Yazılım Altyapısı

Yazılımlarda hazır kütüphane veya üretici SDK'sı kullanılmadı.

Katmanlar

UYGULAMA/FIRMWARE(demo_v06_firm.c)



SÜRÜCÜ (uart.c, i2c.c)



HAL KATMANI (gpio.hal, uart.hal)



DONANIM

Örnek: UART ile tek bir bayt göndermek

```
uart_data_tx("MERHABA")
```



baytı yazmaca koy → başlat → UART'ın bitirmesini bekle



```
0x4000_020C(UART_TDR) = 'M'  
0x4000_0210(UART_CFG)0.biti 1 yap
```



UART birimi baytı bit bit hatta sürer



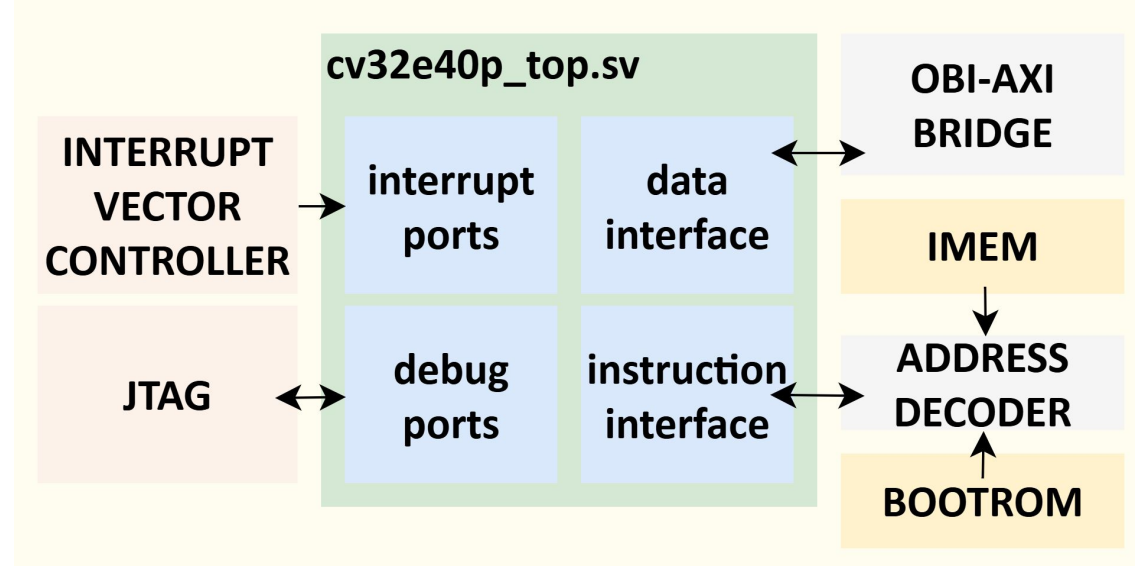
Sürücü Katmanı

Sürücü Uygulamaya sunduğu fonksiyonlar

UART	<code>bd_rt_cfg()</code> baud hızını ayarlar	<code>stp_bt_cfg()</code> stop bitini ayarlar	<code>uart_data_tx()</code> bayt dizisini gönderir	<code>uart_data_rx()</code> enter görene kadar bayt alır, hedefe yazar
I2C	<code>i2c_s_dvc_addr()</code> slave adresini ayarlar	<code>i2c_data_tx()</code> 1-4 bayt yaz	<code>i2c_data_rx()</code> 1-4 bayt oku	
QSPI	<code>qspi_build_ctrl()</code> hangi komut, kaç hat, kaç byte isteği hazırlar	<code>qspi_write_ctrl()</code> hazırlanan istek flasha gönderilir	<code>qspi_wait_done()</code> flash cevabının gelmesi beklenir	
TIMER	<code>timer_init()</code> sayacı kurar, hız vb	<code>timer_enable()</code> saymayı başlatır	<code>timer_get_event_count()</code> event counteri okur	
GPIO	<code>gpio_write_pin()</code> tek pini 1/0 yapar	<code>gpio_generate_pwm()</code> yazılımsal PWM üretir	<code>gpio_measure_frequency()</code> giriş frekansını ölçer	
IVC	<code>ivc_init()</code> kesme altyapısını kurar	<code>ivc_enable()</code> kaynakların interruptlarını almayı aktif eder	<code>ivc_clear()</code> interrupt olayını temizler	
YZ (AI)	<code>ai_init()</code> mod, dolgu ve kesmeyi kurar	<code>ai_wait_done()</code> çıkarmının bitmesini bekler	<code>ai_build_reply_frame()</code> sonucu 8 baytlık cevaba dönüştürür	
OLED	<code>ssd1306_command()</code> ekrana komut gönderir	<code>ssd1306_command_param()</code> parametrelili komut gönderir	<code>ssd1306_data()</code> ekrana gösterilecek verileri yollar	



İşlemci Çekirdeği: CV32E40P

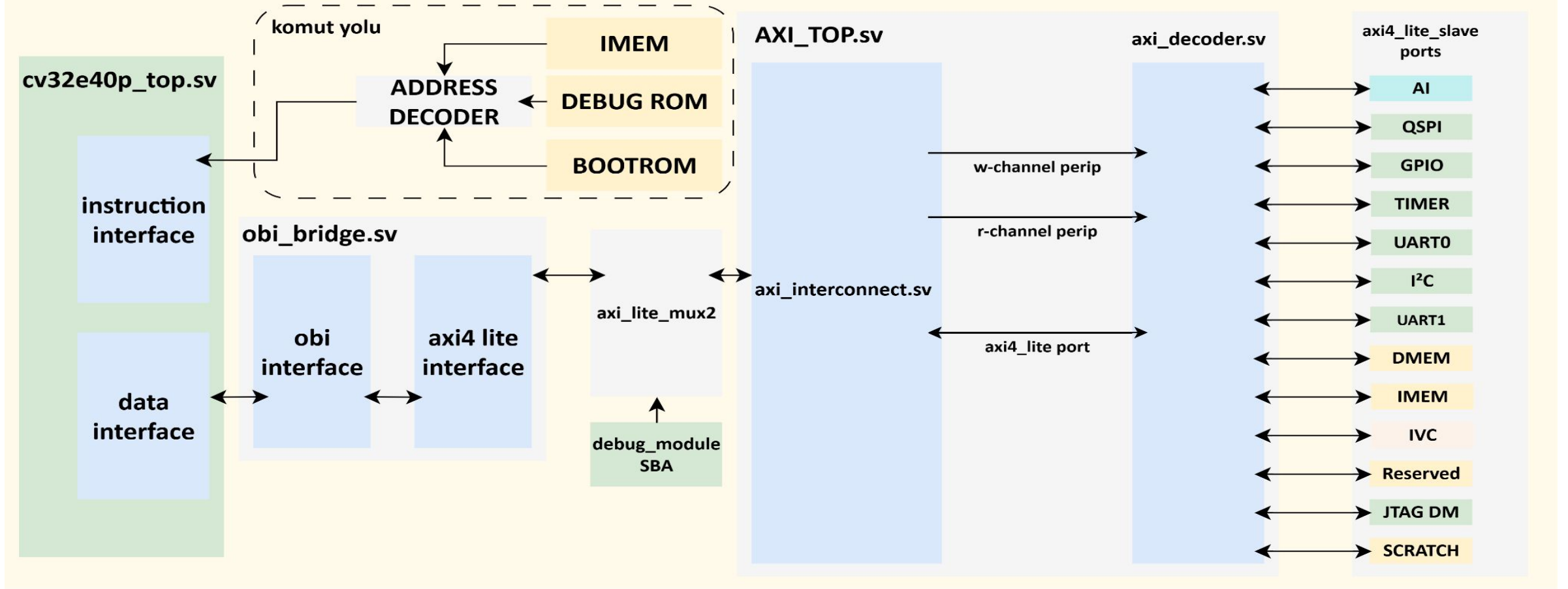


İşlemci portlarının bağlantı yapısı

- **FPU devre dışı:** Hedef uygulama tamsayı tabanlı olduğundan Floating Point Unit çıkarılarak alan ve kaynak tasarrufu sağlandı.
- **Register file → LUTRAM:** FPGA yapısına uyumsuz register file modülü distributed RAM kullanacak şekilde yeniden yapılandırıldı; **~%30 kaynak optimizasyonu.**
- **Port ayrımı:** Data portları obi_bridge üzerinden AXI'ye; instruction portları adress_decoder.sv üzerinden IMEM/BootROM'a yönlendirilir.



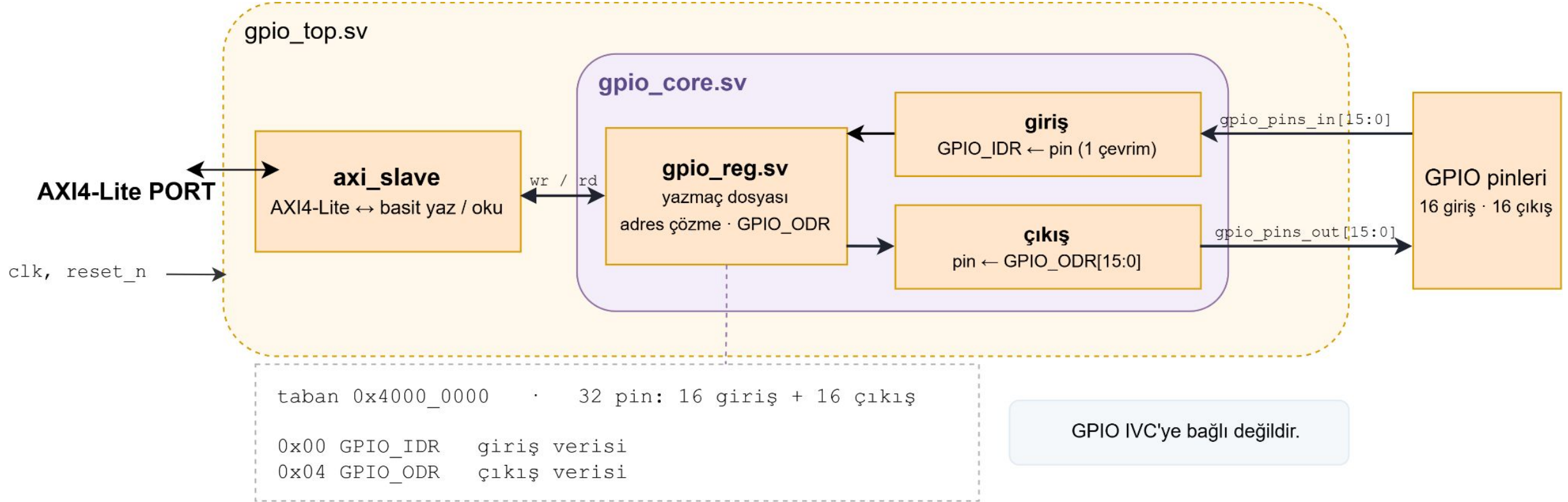
Veri Yolu Mimarisi: OBI → AXI4-Lite



Bus yapısının genel mimarisi

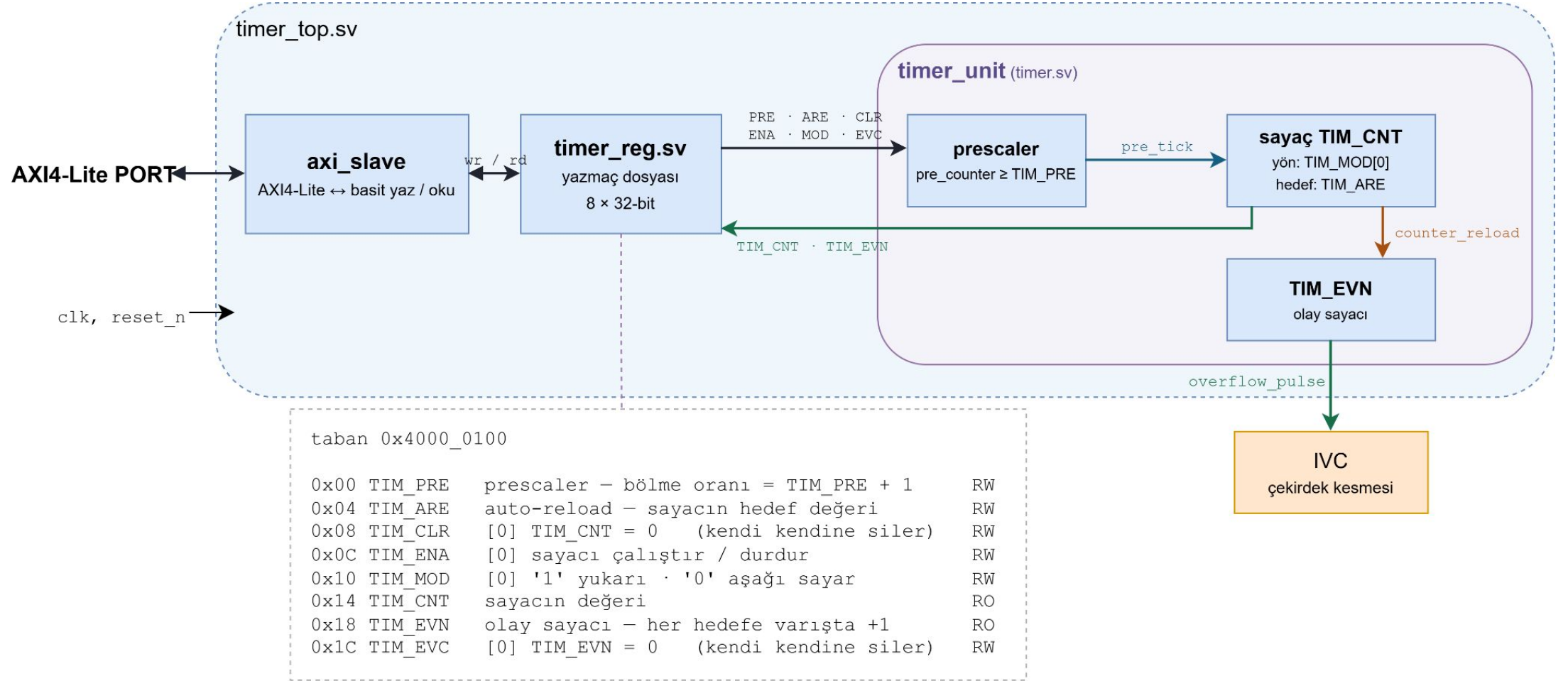


Çevre Birimleri : GPIO



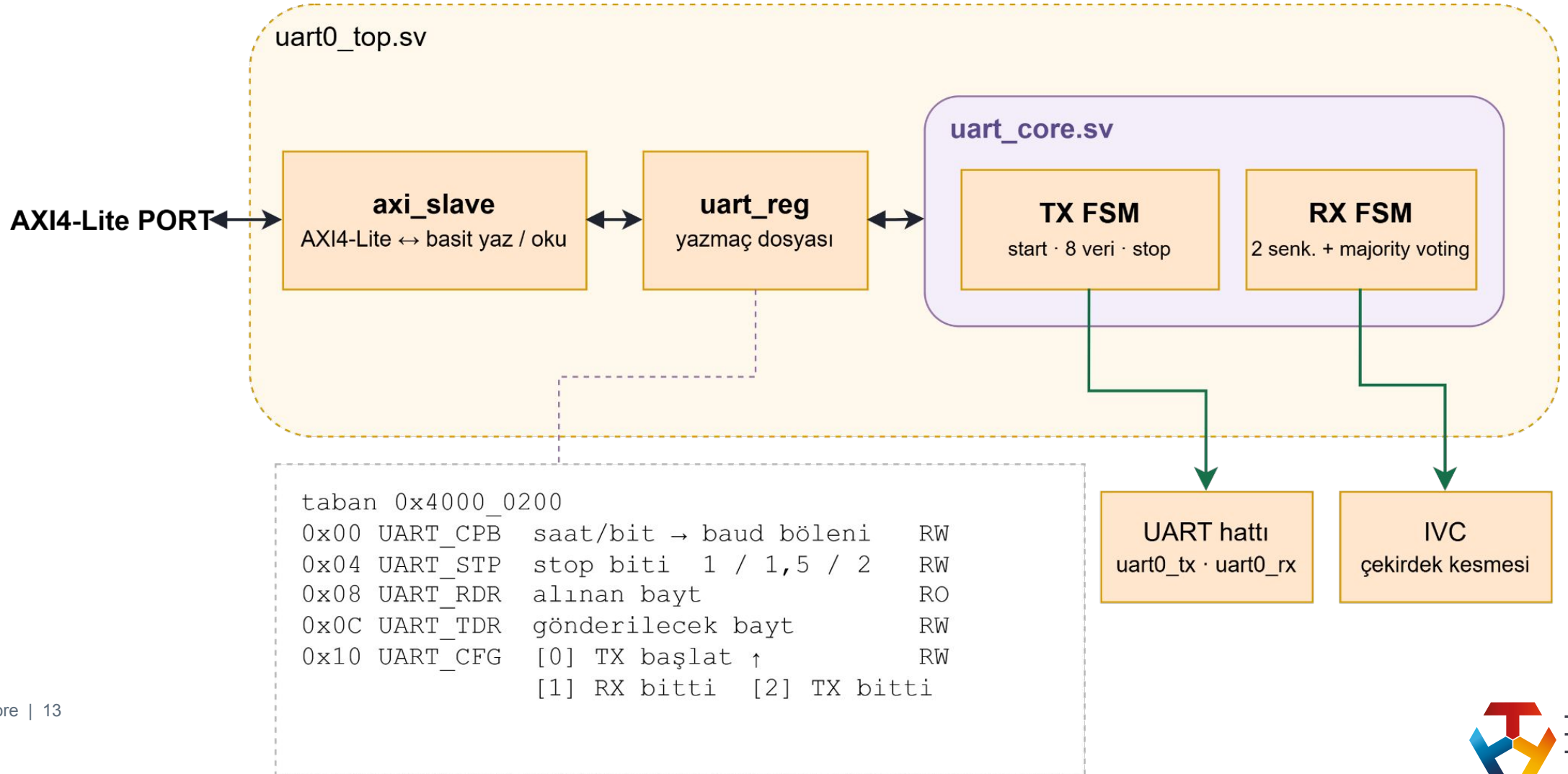


Çevre Birimleri : Timer

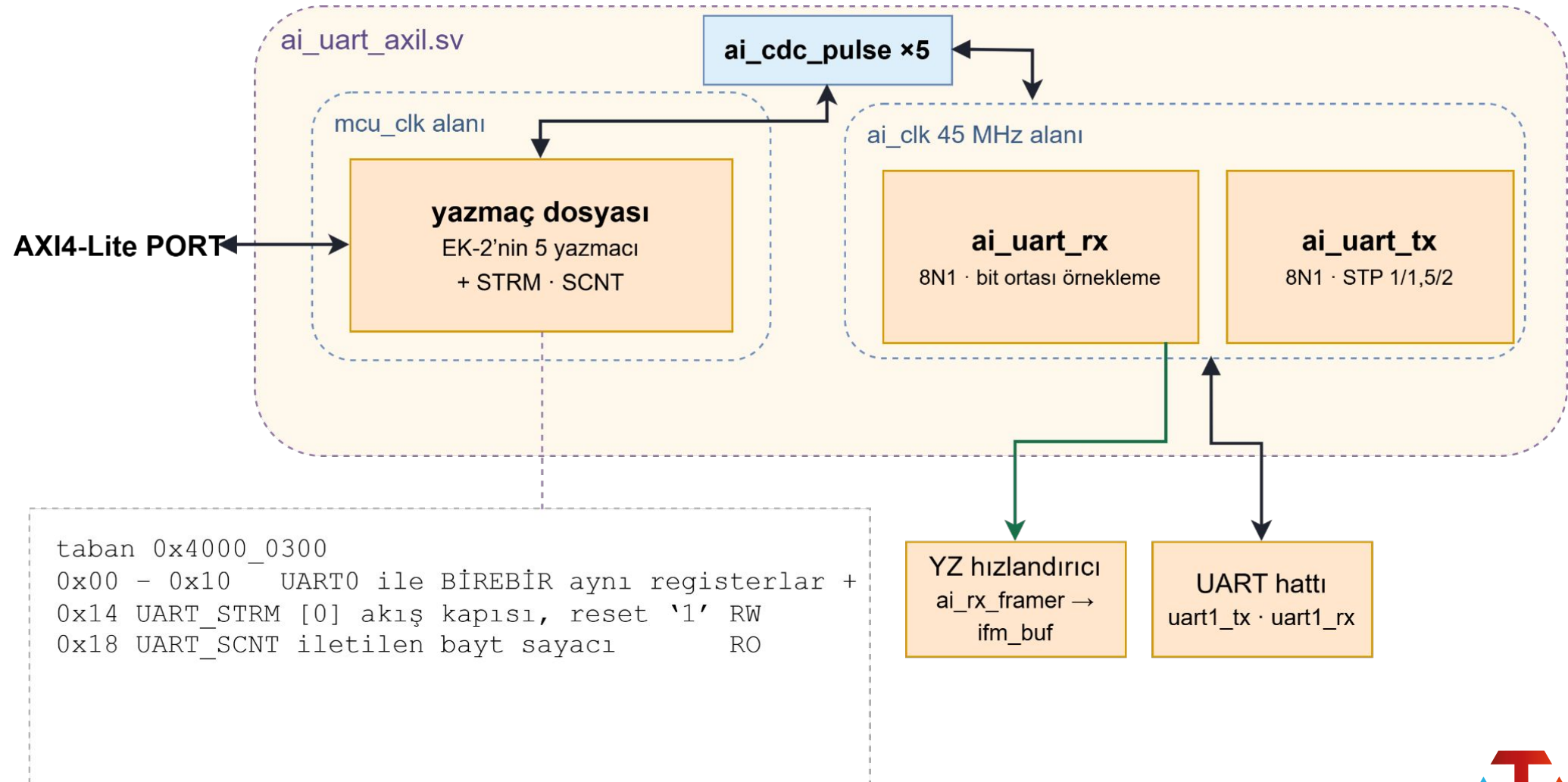




Çevre Birimleri : UART

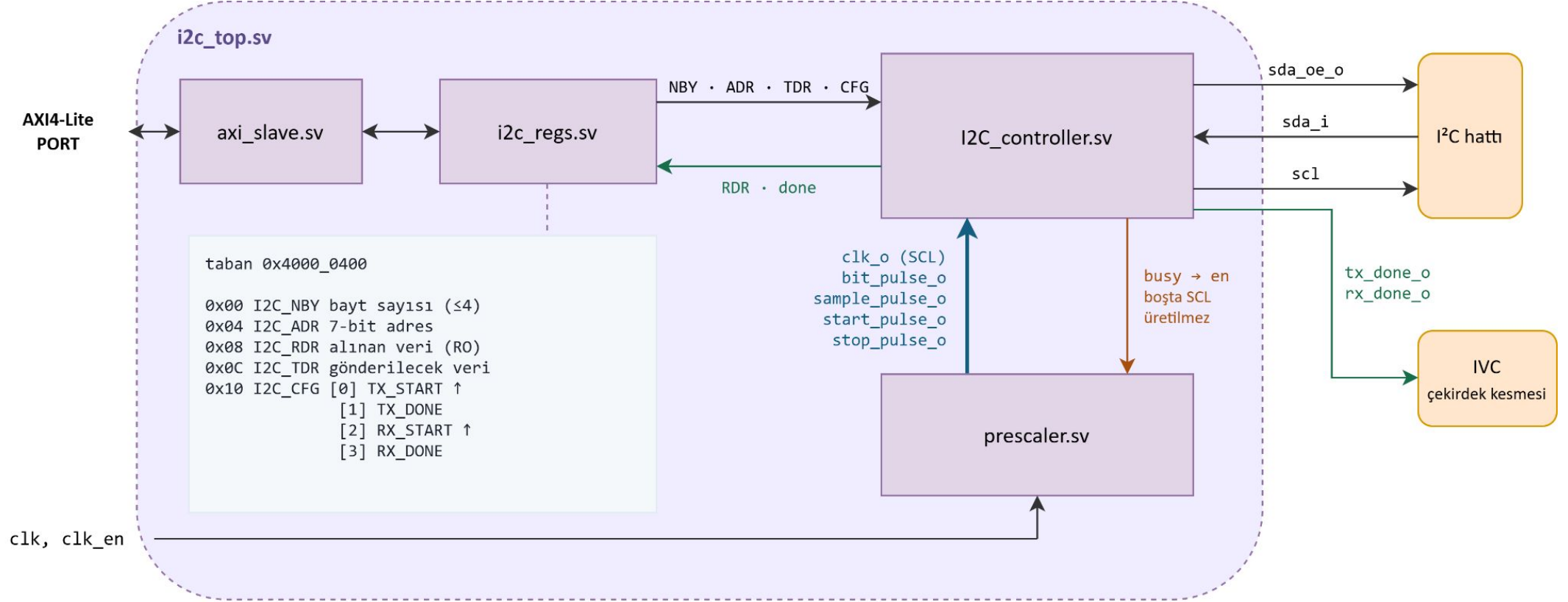


Çevre Birimleri : UART Stream





Çevre Birimleri: I2C



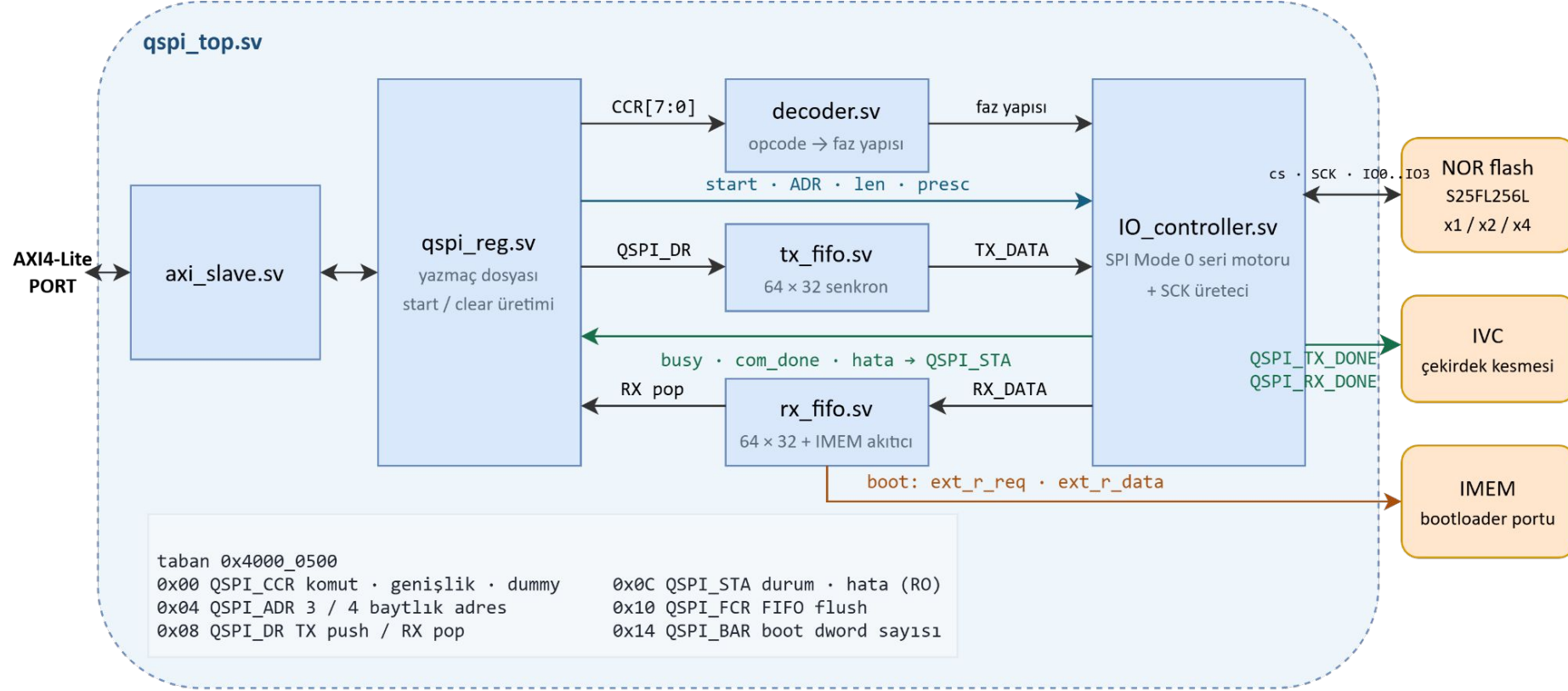
SCL 400 kHz sabit hızda

Adres NACK'inde 3 deneme sonrası işlem iptali

Clock stretching mekanizması bulunmuyor



Çevre Birimleri: QSPI



FPGA ÇALIŞMALARIMIZDA Infineon S25FL256LAGMFI000 NOR FLASH'INI KULLANDIK (DDK ONAYLI)

Şartnamede istenen RESET ve READ_ID komutları flashımızda bulunmadığından o komutları başka bir flash model ile doğruladık.



Boot

Flash'taki program, çalıştırılmak üzere IMEM'e taşınır



8 kB

sabit aktarım · IMEM'in tamamı dolduruluyor

≈8,4 ms

yükleme süresi

4 hat

aynı anda okunur



JTAG (Opsiyonel)

Standart uyumlu

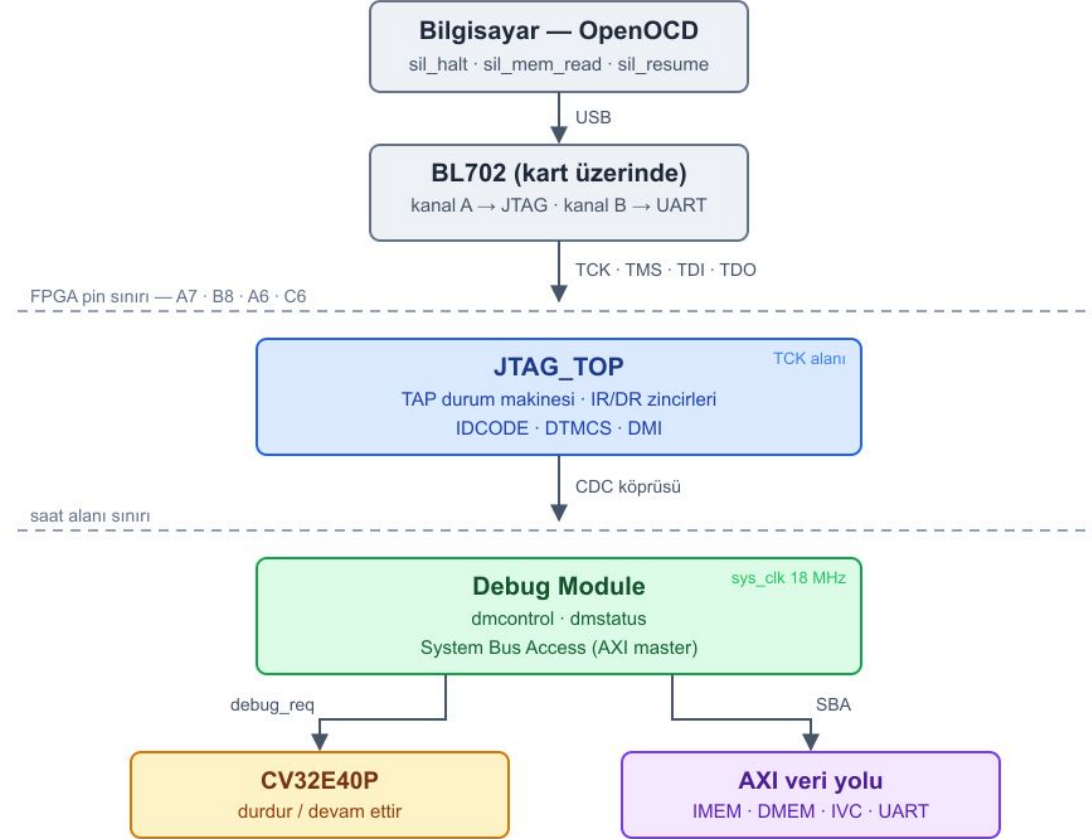
IEEE 1149.1 uyumlu TAP

CPU'ya entegrasyon

Debug Module →
CV32E40P debug_req

Kartta doğruladı

Yandaki Şemadaki
Şekilde Doğrulandı





JTAG (Opsiyonel)

```
Windows PowerShell
--- 2) DTMCS ---
DTMCS = 0x00001071 version=1 abits=7 dmistat=0 idle=1
--- 3) dmcontrol yaz/oku (ALAN SIRASI dogrulaması) ---
--- 4) IMEM icerigi (SBA, islemci calisirken) ---
UYARI: DMI op=02 (0=OK bekleniyordu) -- dmireset atiliyor
MEM[0x00001000] = 0x00000000
KALDI: IMEM okuması yanıt vermiyor (0x00000000)
--- 5) HALT ---
DMSTATUS = 0x00000382 version=2 allhalted=1 allrunning=0 allresumeack=0
CEKIRDEK DURDU
--- 6) IVC pending (0x20020004) ---
MEM[0x20020004] = 0x00000001
--- 7) IVC maskesini ac -> kesme kalkmalı ---
UYARI: DMI op=02 (0=OK bekleniyordu) -- dmireset atiliyor
MEM[0x20020000] <= 0x0000ffff
MEM[0x20020000] = 0x00000041
--- 8) RESUME ---
DMSTATUS = 0x00030c82 version=2 allhalted=0 allrunning=1 allresumeack=1
CEKIRDEK DEVAM EDİYOR
SONUC: 1 KONTROL KALDI
DMSTATUS = 0x00030382 version=2 allhalted=1 allrunning=0 allresumeack=1
CEKIRDEK DURDU
MEM[0x00001000] = 0x20001117
MEM[0x00001004] = 0x00010113
UYARI: DMI op=02 (0=OK bekleniyordu) -- dmireset atiliyor
MEM[0x00001002] = 0x00000000
UYARI: DMI op=02 (0=OK bekleniyordu) -- dmireset atiliyor
UYARI: DMI op=02 (0=OK bekleniyordu) -- dmireset atiliyor
MEM[0x00004000] = 0x00000000
```

Başarılı Testlerimizden Birinin Fotoğrafı
Tang Primer 20K · OpenOCD 0.12

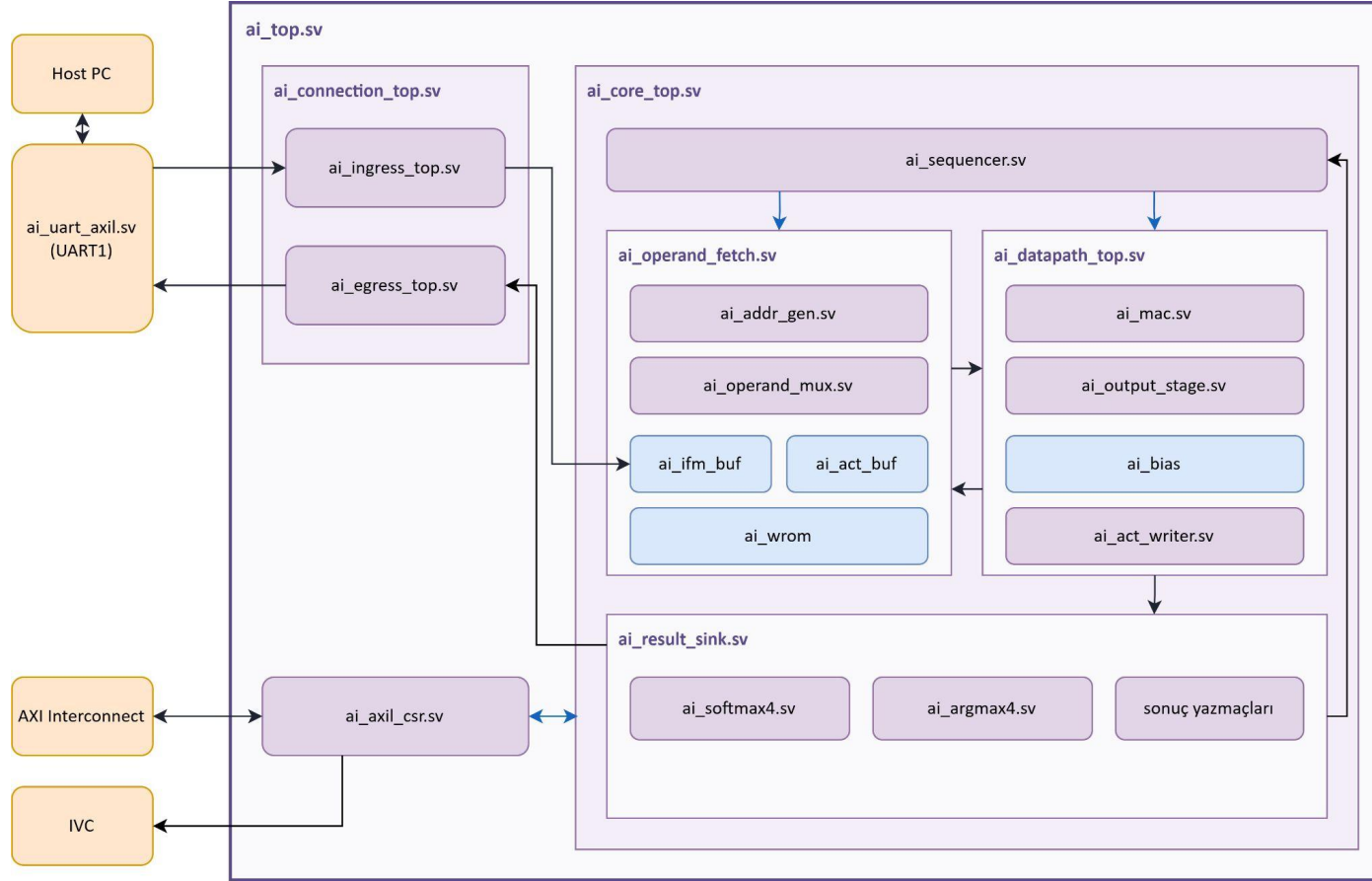


T.C. SANAYİ VE
TEKNOLOJİ BAKANLIĞI



YZ Hızlandırıcı

TFLite Micro Speech "tiny_conv", int8 · tek MAC şeritli gerçek çevrim çekirdeği



45 MHz

ai_clk

336.000

MAC / çıkartım

9,1 ms

ölçülen çıkartım süresi

30 kB

hızlandırıcı belleği

11.25 MHz

ai_rom_clk



YZ Hızlandırıcı: Protokol ve Yazmaç Arayüzü

UART üzerinden veri, AXI4-Lite üzerinden kontrol

RX Frame bilgisayar → kart



XOR bunları kapsar — sync hariç, uzunluk dahil

Uzunluk çerçevesinin içinde: N = 1960 (FULL · 49×40 görüntü) veya 40 (SLIDING · tek dilim). Tek bitstream ikisini de kabul eder.

TX Frame kart → bilgisayar · sabit 8 bayt



stat = seq << 2 | sınıf → sınıf, stat'ın alt 2 biti

VERSION[27:24] “datapath_real”: çalışan bitstream'in gerçek çekirdekle mi yoksa zamanlama stub'ıyla mı derlendiğini tek okumada söyler aynı zamaanda eski bir bitstream'i tespit etmenin yoludur.

Kategori	Yazmaç	Amaç
Kontrol	CTRL · IRQ_EN	çıkartım başlat, kesme etkinleştir
Durum	STATUS · VERSION	busy/done/hata, bitstream kimliği
Sonuç	CLASS · LOGITS · CYCLES	sınıf, 4×int8 logit, ölçülen süre
Konfigürasyon	PAD_VALUE · MODE · DUMMY_DELAY	dolgu değeri, FULL/SLIDING seçimi
Sayaç	RX_FRAMES · TX_FRAMES · RX_ERRORS	çerçeve ve hata sayaçları

13

yazmaç alanı

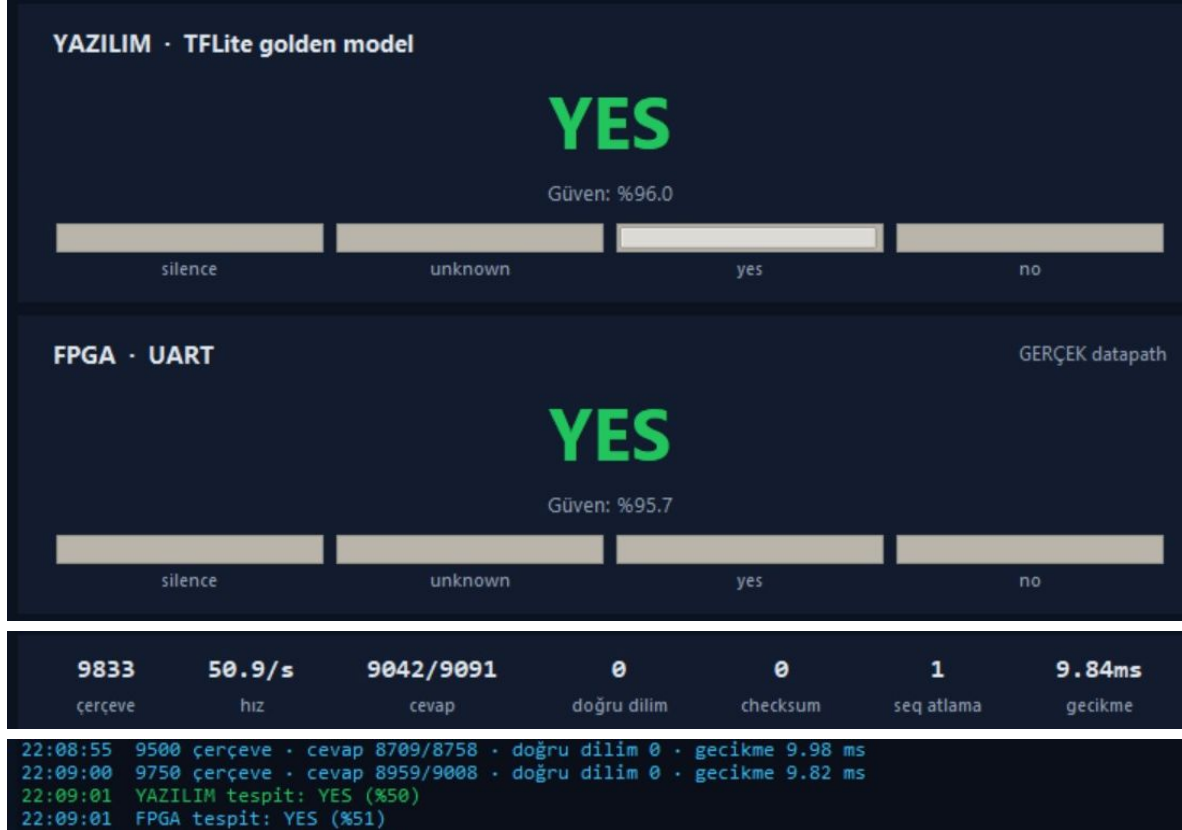
0x3000_0000

taban adres (AI CSR)



YZ Hızlandırıcı: Altın Model ile Doğrulama

Aynı ses iki bağımsız yoldan geçiyor — üstte yazılımın cevabı, altında FPGA'nın cevabı



Kanıtın mantığı

Ses çerçevesi
20 ms



TFLite altın modeli
bilgisayarda

YES
%96,0

aynı sınıf · 0,3 puan



Hızlandırıcı
UART · FPGA

YES
%95,7



Neden Sadece Dosya Değil, Canlı Mod da Var

Nihai doğrulaması tek seferlik dosya gönderimi + modelin kendi sliding window çalışma prensibi

Dosya Modu Veri Akışı

Pencere her seferinde sıfırdan kurulur: 1 saniye dinle, tamamını gönder, bir kez çıkarım yap.



Çıkarım anı — 1 saniyede 1 kez



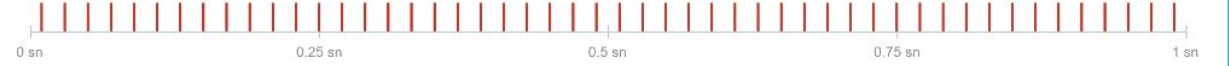
Veri: 1960 B/s · Gecikme: bir sonuç için tam 1 saniye beklemek gerekir.

Sliding Window Veri Akışı

Pencereyi FPGA tutar: PC yalnızca yeni dilimi yollar, halka kayar, her dilimde bir çıkarım.



Çıkarım anı — 1 saniyede ~50 kez



Veri: 2000 B/s — v1 ile neredeyse aynı. 50 kat sonuç, aynı bant genişliği: pencerenin 48/49'u FPGA'da duruyor, tekrar gönderilmiyor.

Arka arkaya iki kelime

0,5 sn arayla

- Mod 1 1 kelime yakaladı
- Mod 2 2 kelimeyi de yakaladı

Kısık ses

gain 0,02

- Mod 1 hiç tetiklemedi
- Mod 2 49/101 pencerede doğru

Doğrulama noktası

saniyede

- Mod 1 1 karşılaştırma
- Mod 2 ~50 karşılaştırma

Mod 1 ve Mod 2 arası geçişler YZ Hızlandırıcı donanımı tarafından otomatik bir şekilde gerçekleştirilir



Canlı Modun Donanım Maliyeti

Kayan pencere modu eklendi — hesaplama mantığı hiç değişmedi

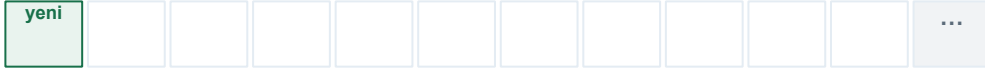
Ne değişti

ÖNCE · FULL her çıkartım için tüm görüntü
1960 bayt = 49 satır × 40 · telde 6,55 ms

SONRA · SLIDING her 20 ms'te yalnızca YENİ dilim
40 bayt = 1 satır · telde 150 µs

Peki eski 48 satır nereden geliyor?

Donanım onları zaten tutuyor



base en eski satırı gösterir; yeni dilim onun ÜSTÜNE yazılır ve base bir ilerler.

Okuyucu $r = (iy + base) \bmod 49$ yapar — 49 satır = 1 saniyelik pencere.

Bunun donanım maliyeti

- Çerçeveye 2 baytlık uzunluk alanı**
Mod artık derleme parametresi değil, çerçevenin içinde: 40 → dilim, 1960 → tam görüntü. Tek bitstream ikisini de kabul eder.
- ifm_buf ring buffer oldu**
Satırlar yerinde kalıyor, yalnızca okuma adresi döndürülüyor — veri kopyalanmıyor.
- Dilim için 40 baytlık ara kayıt**
Görüntü doğrudan buffera akar; dilim ise sağlaması tutana kadar ara kayıta bekler — bozuk bir paket birikmiş 1 sn'yi bozmasın diye.

320

flip-flop

%46

20 ms bütçenin kullanımı



YZ Hızlandırıcı: Testbench'ler

Blok seviyesinde ve uçtan uca testler

Üç seviye · modelin her katmanı ayrı sinanıyor

Blok seviyesi

Çarpma-toplama, bias + ReLU, softmax, argmax, ağırlık ROM'u, adres üretici, bufferlar

tb_ai_mac · tb_ai_bias · tb_ai_softmax4_top · tb_ai_argmax4 · tb_ai_wrom · tb_ai_addr_gen · tb_ai_bufs

Alt sistem

UART çerçevesinden belleğe (ingress), çekirdek veri yolu, sonucun geri çerçevelenmesi (egress) ve yazmaç arayüzü.

tb_ai_ingress_top · tb_ai_datapath_top · tb_ai_egress_top · tb_ai_operand_fetch · tb_ai_axil_csr

Uçtan uca

CPU'suz tam hızlandırıcı; ayrıca sistem tezgâhında CPU → CSR → çıkartım → kesme → UART zincirinin tamamı.

tb_ai_top · tb_ai_golden · tb_demo1_0 (T14)

Altın Vektör Testi

Aynı öznitelik haritası hem referans modelden hem RTL'den geçiyor; kazanan sınıf ve dört olasılık baytı **bit-bit** karşılaştırılıyor.

ramp

adres üreticide off-by-one

min / max

int8 uçları, doyum

lfsr

yapısız veri

checker

yüksek uzamsal frekans



FPGA Prototipleme

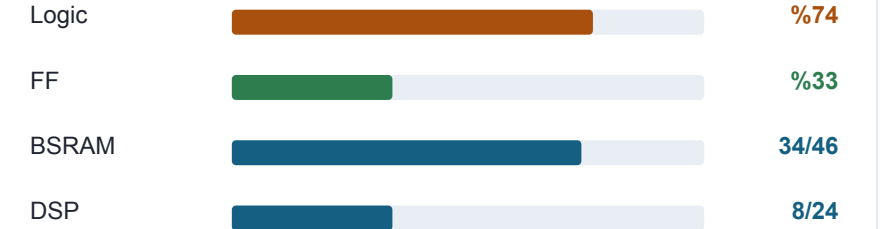
Çevre birimleri tek tek eklendi — her basamak kartta doğrulandı

- 1 demo v0.1 TIMER
- 2 demo v0.2 IMEM · adres decoder · GPIO · QSPI
- 3 demo v0.4 DMEM · UART0 · I2C
- 4 demo v0.6 JTAG (TAP) · IVC · AXI MUX
- 5 demo v1.0 UART1 akışı · YZ hızlandırıcı



I2C ile sürülen OLED — demo v04 basamağı, kart üzerinde

GOWIN GW2A-18 · nihai tasarımın doluluk





Doğrulama

EK-3'teki üç zorunlu maddenin tamamı karşılandı.

verification/

```
├ peripherals/  gpio timer uart
|               i2c qspi jtag
├ cpu/         core + spike_iss
├ system/      boot, uçtan uca demo
├ axi/         VIP + directed + memory
├ ai/          YZ hızlandırıcı testleri
├ uvm/         AXI4-Lite UVM ortamı
├ support/     sim modelleri, araçlar
└ docs/        test planları + coverage
```

EK-3 Doğrulama Metodu	Öncelik
Doğrulama Planı	Elden Gelenin En iyisi
Blok Seviyesi Testler	Opsiyonel
Protokol Kontrolleri	Zorunlu
Çekirdek Testleri	Elden Gelenin En iyisi
YZ Hızlandırıcı Testleri	Zorunlu
Sistem Seviyesi Testler	Zorunlu
Code Coverage	Opsiyonel
Functional Coverage	Opsiyonel



Doğrulama Planı

Yürütülen doğrulama faaliyetlerinin ana hatları ve tamamlanma hedefleri için ayrı test planı dosyaları

Birim	Dosya
GPIO	gpio_test_plan.xlsx
I2C	i2c_test_plan.xlsx
QSPI	qspi_test_plan.xlsx
Timer	timer_test_plan.xlsx
UART	uart_test_plan.xlsx
Çekirdek (Spike)	spike_test_plan.xlsx
AXI4-Lite (UVM)	axi_uvm_test_plan.xlsx

i2c_test_plan.xlsx

Feature	Verification Goals	Property or Checker	Type	Reason
Reset Check	Reset durumunda tüm registerlar 0 mı?	R-01	RTL	reset durumunda yazmaçların temizlenmesi
Simultaneous TX/RX Protection	Aynı anda hem TX hem RX tetiklendiğinde TX işlemi önceliklendiriliyor mu?	R-02	RTL	şarname gereği eşzamanlı TX/RX çıkışmasında donanımın TX'e öncelik vermesi
NBY Rounding & Boundary	NBY yazmacına 1-4 dışındaki geçersiz değerler yazıldığında yuvarlama yapılıyor mu?	R-03	RTL	şarname gereği geçersiz NBY değerlerinin 1-4 aralığına yuvarlanması

Şablon: OpenHW Group CORE-V Formal VerifPlan şablonundan yararlanıldı.



Çevre Birimleri Doğrulaması

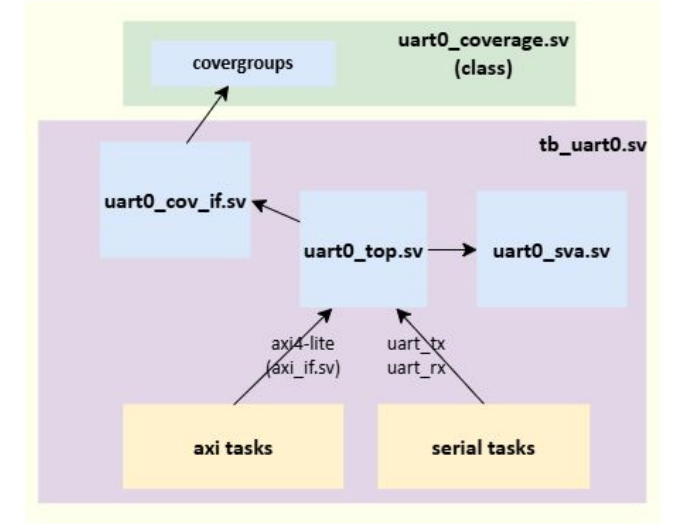
Her çevre birimi için aynı dört parçalı, tekrarlanabilir doğrulama ortamı

- 1 **tb_<birim>.sv**
En üst seviye testbench: saat üretimi, AXI4-Lite ve seri hat görevleri, yönlendirilmiş testler
- 2 **<birim>_cov_if.sv**
RTL iç sinyallerini (tx_state, rx_state, UART_CPB...) assign ile kapsam ortamına taşır.
- 3 **<birim>_coverage.sv**
Class tabanlı covergroup'ları içerir. Şartname isterlerini doğrudan bin'e çevirir (örnek: 1 Mbps → CPB = 18).
- 4 **<birim>_sva.sv**
R (tasarım), I (protokol), P (performans) kategorili assertion'lar simülasyon boyunca kuralları denetler.

uart0_sva.sv : 282-287

```
// [R-09b] EK-2 v1.3: "gönderim tamamlandığında bu bit lojik devrenin
// kendisi tarafından '0'a çekilmelidir"
property p_tx_done_clears_start;
  @(posedge clk) disable iff (!reset_n)
    tx_done_tick | => (UART_CFG[0] == 1'b0);
endproperty
a_tx_done_clears_start : assert property (p_tx_done_clears_start)
  else begin $error("[R-09b] EK-2 ihlali: CFG[0] donanım tarafından indirilmedi!"); note("R-09b"); end
```

uart0 çevre birimi test mimarisi





Çevre Birimleri Doğrulaması

Ayrıca Çeşitli, Bağımsız IP'lerle de testler yapıldı.

OpenCores I2C_SLAVE_MODEL

I2C

opencores.org referans slave modeli
(R. Herveille / J. Sheahan)

I2C protokol uygunluğu doğrulandı

Micron MT25QL256ABA

QSPI

Şartnamenin istediği referans parça;
üreticinin resmî Verilog modeli

Şartnamedeki Parça ile uyumluluk doğrulandı

N25Qxxx.v davranış modeliyle

Infineon S25FL256SAGMFI000

QSPI

Fiziksel Flashımızın içermediği iki komut:

READ_ID ve RESET komutları doğrulandı

Infineon S25FL256LAGMFI000

QSPI

Kartta takılı olan gerçek parça;
üreticinin resmî Verilog modeli

**READ_ID yerine RDID ve RES
RESET yerine RSTEN ve RSTCMD
komutları kullanıldı**



Çevre Birimleri Doğrulaması

uart0 ve i2c cocotb repoları ile python tabanlı testbenchlerle doğrulandı

yaptığımız python tabanlı testbenchler

UART0

cocotbext-uart repolu ile

uart_core doğrudan sürülüyor; bağımsız
UartSource / UartSink modelleri

TX ve RX test edildi

test_uart_core.py

I2C — tam AXI zinciri

**cocotbext-axi + cocotbext-i2c repoları
ile**

Bağımsız AXI4-Lite master ve I²C slave modelleri ile

Axi- RTL - i2c hattı test edildi

test_i2c_master.py

I2C — yazmaç katmanı

cocotbext-i2c ile izole i2c_reg testi

Bazı şartname isterleri doğrulandı Örn: NBY'ye 25
yazıldığında 4'e yuvarlanması gerektiği gibi

Bazı şartname isterleri test edildi.

test_i2c_reg.py



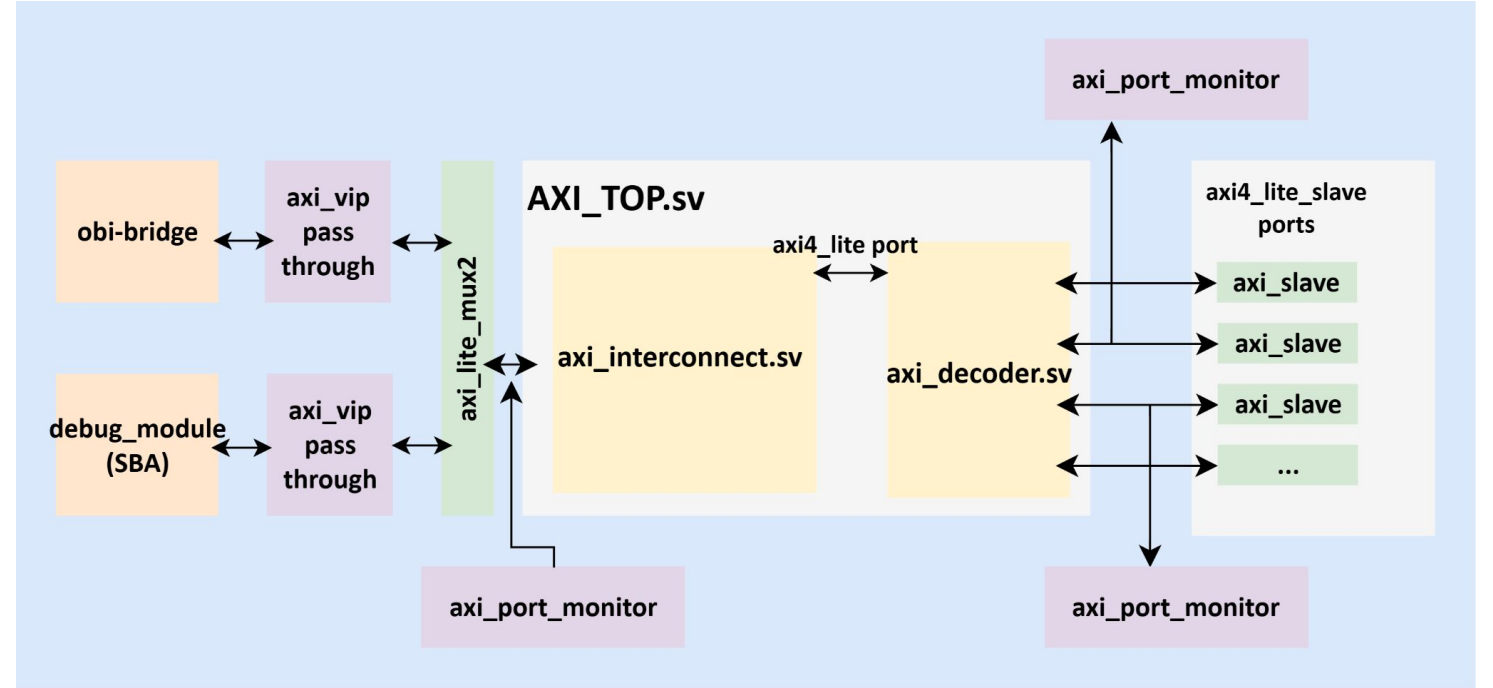
AXI VIP ve Protocol Checker

Simülasyon boyunca protokole aykırı bir davranışı aktif olarak yakalama

AXI VIP : Xilinx IP kataloğundan bir Verification IP

AXI Protocol Checker : Arm'ın AMBA
referans denetleyici Xilinx IP

2 Farklı Testbench Kullanıldı

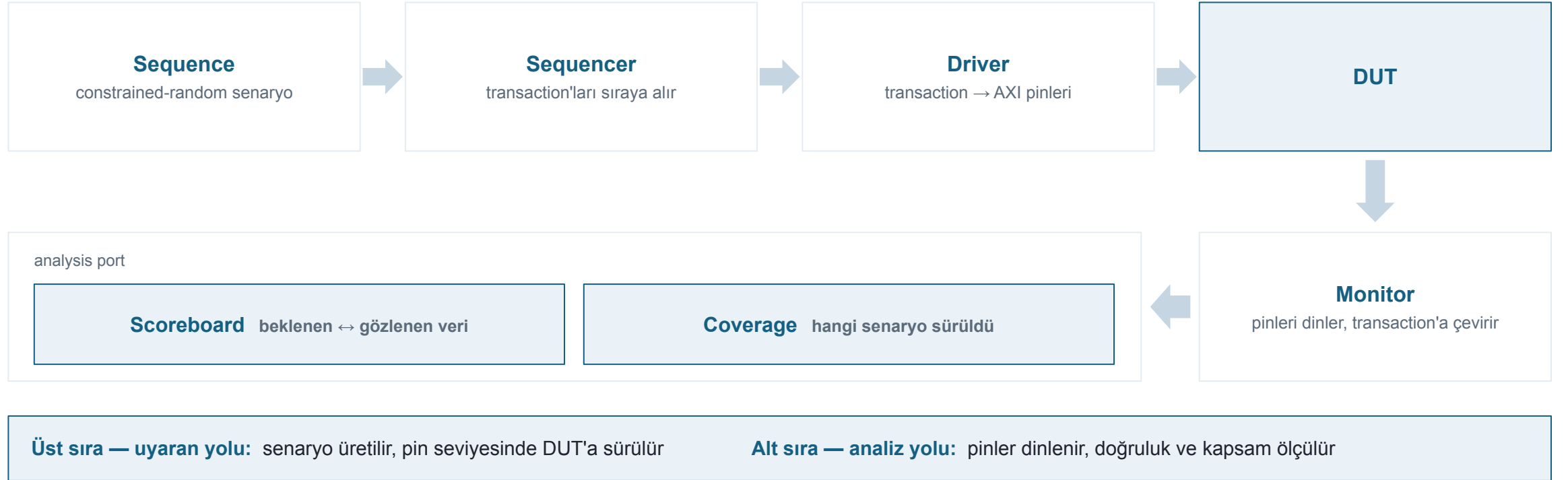


axi vip ve protocol checker doğrulama ortamı



UVM doğrulama ortamı

Testbench mimarisi





UVM doğrulama ortamı

Bus Yapısı

UVM agent
driver · monitor

AXI_TOP

interconnect

decoder



13 slave — bellek modeli, her portta SVA

İşlem doğru slave'e gidiyor mu?

Bu testte slaveler bellek modeli

Slaveler + mux

UVM agent
driver · monitor

gpio_top
gerçek RTL

×12 tezgah: gpio · timer · uart0 · uart1 · i2c · qspi ·
debug_module
+ imem · dmem · IVC · null_slave · mux2 (iki agent)
CPU yok, interconnect yok

Her AXI Slave'i tek başına protokole uyuyor mu?

her biri **yalnız başına**, UVM agent'ı doğrudan onun AXI portuna bağlı.

YZ Hızlandırıcı

UVM agent
driver · monitor

ai_axil_csr
AXI4-Lite yazmaç arayüzü

axi_clk

ai_clk

hızlandırıcı çekirdeği

BFM — gerçek yerine

CDC

CDC AXI'yi kilitliyor mu?

ai_clk durursa config yazması asla bitmez — tek başına ancak burada sürülür

YZ Hızlandırıcının AXI'ye bakan iki arayüzü de UVM ile sürülüyor



UVM doğrulama ortamı

UVM functional coverage — üç covergroup ne ölçüyor

axi_lite_cg

Omurga · DUT: AXI_TOP

13 slave portunun her birine hem okuma hem yazma yapıldı mı?

cp_slave hangi köle: 13 yuva + haritasız
cp_cmd okuma / yazma
cp_resp cevap: okay · slverr
cp_window adres pencere içi / dışı
cp_strb hangi baytlar yazıldı: 8 desen
cp_align hizalı / hizasız adres

cross: cmd × slave
+ cmd×resp · cmd×window · strb×cmd

csr_axi_cg

YZ CSR · DUT: ai_axil_csr

13 yazmacın hepsine hem okuma hem yazma yapıldı mı?

cp_reg 13 yazmacın her biri + haritasız adres
cp_cmd okuma / yazma
cp_strb 16 bayt-şerit kombinasyonu
cp_align hizalı / hizasız adres

cross: reg × cmd
+ strb×cmd

csr_status_cg

YZ CSR · yalnızca STATUS okumasında örneklenir

STATUS'un 5 biti hem 0 hem 1 görüldü mü?

cp_busy meşgul · 0 ve 1
cp_done bitti · 0 ve 1
cp_irq kesme · 0 ve 1
cp_crc CRC hatası · 0 ve 1
cp_ferr çerçeve hatası · 0 ve 1

Neden var: bu 5 bit AXI'den yazılmaz, ai_clk'ten CDC'yi aşarak gelir — hepsini hem 0 hem 1 görmek, olay yolunun gerçekten sürüldüğünün kanıtıdır.



Çekirdek Testleri

Spike ISS ile RTL arasında offline, self-checking karşılaştırma yaptık

Buyruk kümesi	Kapsanan	Oran
RV32I — tüm gruplar (aritmetik, dallanma, yükleme/saklama, sistem)	41 / 41	%100
Zicsr	6 / 6	%100
RV32M — çarpma / bölme	8 / 8	%100
RV32C — sıkıştırılmış	18 / 18	%100

```
verification/cpu/spike_iss
```

```
└ basic_isa_tests/  
└ directed_functional_tests/  
└ riscv_tests/
```

Tüm instructionları test edebilmek
için **riscv-tests** deposu kullanıldı.

`bubble_sort`, `matrix_multiplication`
gibi test senaryoları çalıştırıldı.



Code Coverage

Modül bazında statement, branch and condition coverage'lar

GPIO

gpio_core

Statement		%100
Branch		%100
Condition		%100

Timer

timer_unit

Statement		%100
Branch		%100
Condition		%100

UART

uart_core

Statement		%100
Branch		%92.7
Condition		%97.5

UART-stream

ai_uart_axil

Statement		%100
Branch		%86.3
Condition		%81.3

I2C

i2c_controller

Statement		%99.2
Branch		%95.6
Condition		%100

QSPI

IO_controller

Statement		%95.5
Branch		%87.9
Condition		%98.6



Code Coverage



Dashboard Files Modules

File Summary of C:/Users/Arda/Documents/GitHub/2026_teknofest_Silicore/rtl/peripherals/I2C/prescaler.sv

File ID	Modules Count	Total Instances Count	Total Lines	Total Statements	Statement Coverage Score	Show Statement Coverage	Branch Coverage Score	Show Branch Coverage	Condition Coverage Score	Show Condition Coverage	Toggle Coverage Score	Show Toggle Coverage
5	1	1	22	22	100	☒	100	☐	100	☐	0	☐

Statement Coverage of File 5 : C:/Users/Arda/Documents/GitHub/2026_teknofest_Silicore/rtl/peripherals/I2C/prescaler.sv

Statement Coverage Summary for File : C:/Users/Arda/Documents/GitHub/2026_teknofest_Silicore/rtl/peripherals/I2C/prescaler.sv

Total Statements	Covered Statements	Uncovered Statements	Statement Coverage Score
22	22	0	100

LineNumber	LineContent	Execution Count	Num Of Statements
1	////////////////////////////////////		
2	// Copyright 2026 Silicore Team		
3	// Author(s): Eyüp Ertan		
4	// Version: v0.4		
5	////////////////////////////////////		
6			
7	module prescaler #(		
8	parameter prescaler_bit_width = 8		
9	)(		
10	input logic clk,		
11	input logic reset_n,		
12	input logic [prescaler_bit_width-1:0] prescaler_i,		



Functional Coverage

Kullanıcı tanımlı covergroup'larımız

Kaynak başına fonksiyonel kapsam

GPIO	3 covergroup		%100
Timer	4 covergroup		%100
UART	4 covergroup		%100
UART-stream	6 covergroup		%100
I2C	2 covergroup		%100
QSPI	2 covergroup		%100



Çevre Birimleri için Oluşturduğumuz 21 Covergroup

Birim	Covergroup'lar			
GPIO 3 covergroup	cg_gpio_access her adrese okuma yazma denendi mi?	cg_gpio_wstrb 16 WSTRB byte-lane kontrolü	cg_gpio_data giriş ve çıkış pin veri desenler (0000,FFFF...)	
Timer 4 covergroup	cg_timer_registers TIM_PRE / ARE / ENA / MOD değerleri	cg_timer_control TIM_CLR, TIM_EVC cidden tetiklendi mi?	cg_timer_events prescaler tik, reload, taşma, olay sırası gözetildi mi?	cg_timer_cross mod × enable × reload × prescaler/aşağı yukarı yön
UART 4 covergroup	cg_uart_registers CPB / STP / TDR / RDR / CFG değerleri	cg_uart_tx_fsm TX durumları ve durum geçişleri	cg_uart_rx_fsm RX durumları ve durum geçişleri	cg_uart_cross stop bit × TX durumu × RX durumu
UART-stream 6 covergroup	cg_uart1_registers CPB / STP / TDR / RDR / CFG + STRM değerleri	cg_uart1_cdc cfg_busy ve tx_busy ölçümü cdc için	cg_uart1_tx_fsm TX durumları ve durum geçişleri	
	cg_uart1_rx_fsm RX durumları ve durum geçişleri	cg_uart1_stream stream_en × RX durumu	cg_uart1_strm_switch stream_en değişikliği 1→0→1	
I2C 2 covergroup	cg_i2c_registers NBY, ADR, CFG TX/RX	cg_i2c_cross NBY × TX/RX		
QSPI 2 covergroup	cg_qspi_registers opcode · genişlik · adres · dummy · prescaler	cg_qspi_phases faz kombinasyonları × genişlik × yön		

Kaynak dosyalar: <birim>_coverage.sv · birleşik rapor: docs/coverage/fc_report_html_merged/ · 21 covergroup



Sistem Seviyesi Testler

Firmware'in SoC üzerinde koştuğu test senaryoları

demo 0.4

Çevre birimi zinciri

+ DMEM · UART0 · I2C

- 1 BootROM flash'tan IMEM'e kopyalar
- 2 UART0'dan açılış mesajı gönderilir
- 3 UART0'dan gelen komut beklenir
- 4 I2C ile SSD1306 OLED başlatılır ve yazılır

demo 0.6

Kesme ve hata ayıklama

+ JTAG · IVC

- 1 BootROM flash'tan IMEM'e kopyalar
- 2 UART0'dan "V06" başlığı gider
- 3 TIMER kesme üretir, IVC CPU'ya taşır
- 4 ISR sayar, sonucu UART0'dan basar
- 5 JTAG çekirdeği durdurur, SBA ile yazmaç okunur, devam ettirilir

demo 1.0

Nihai tasarım

+ UART1 akış · YZ hızlandırıcı

- 1 v06 senaryosunun tamamı
- 2 UART1'den ses vektörü gelir
- 3 Hızlandırıcı çıkarımı yapar
- 4 Kesme → ISR cevabı hazırlar
- 5 UART1'den sınıf ve olasılık gönderilir

```
cd <depo root> source verification/systest.tcl  
systest demo4 demo6 demo10 (Vivado Tcl Console)
```



Sistem Seviyesi Testler

Uçtan uca senaryolar

Birden çok birim aynı zincirde

- demo10** boot QSPI→IMEM, UART0, JTAG/DM/SBA, IVC+Timer, YZ
- demo6** UART0 + TIMER→IVC kesmesi + JTAG/DM/SBA
- demo4** UART0 → I2C/OLED, SSD1306 init dizisi birebir

Çevre birimi sürücü testleri

Tek birim için sistem seviyesi testler

- gpio** PWM periyot ve görev oranı, frekans
- qspi** gerçek S25FL256L flash modeline karşı
- i2c** 400 kHz SCL, EK-2 bayt sırası, NBY
- timer** (ARE+1)×(PRE+1) periyodu, ENA/CLR/EVC
- uart** UART_CPB baud, RX yolu, CFG

Tüm testler self-checking yapıdadır

Kanıt log'u oluşturulur

Ayrıca systest gls

```
cd <depo root> source verification/systest.tcl  
systest all (Vivado Tcl Console)
```



Çip Tasarım Akışı: Sentez ve Zamanlama

SkyWater sky130A · standart hücre kütüphanesi sky130_fd_sc_hs · OpenLane akışı

71,4 MHz

en kötü köşede kapanan frekans

+0,33 ns

hold slack'i

109,8 mW

toplam güç · nominal köşe

8,32 mm²

çip alanı

Zamanlama · en kötü şartlarda

max_ss_100C_1v60

yavaş silikon · 100 °C · 1,60 V

14 ns

saat periyodu

71,4 MHz

karşılığı frekans

Setup WNS

+0,39 ns

Hold WNS

+0,33 ns

Güç · ideal şartlarda

nom_tt_025C_1v80

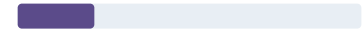
tipik silikon · 25 °C · 1,80 V — normal çalışma şartları

Makrolar · %49,7



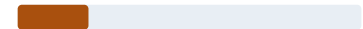
54,6 mW

Saat ağacı · %21,9



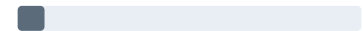
24,1 mW

Ardışıl (DFF) · %20,7



22,7 mW

Kombinasyonel · %7,7



8,4 mW

Toplam

109,8 mW



Çip Tasarım Akışı: Alan ve Hücre Dağılımı

Die 3172,4 × 2622,7 µm · toplam 359.902 yerleştirilmiş hücre + 20 bellek makrosu

8,32 mm²

die alanı

359.902

toplam hücre

%19,0

mantık hücresi oranı

20

SRAM / ROM makrosu

Hücre dağılımı

Fill · 208.772		%58,0
Combinational Logic · 47.887		%13,3
Decap · 44.047		%12,2
Tap Cell · 36.983		%10,3
Buffer · 12.315		%3,4
DFF · 8.065		%2,2
Diode · 1.786		%0,5
Tie Cell · 47		<%0,1

Mantık hücreleri · 68.267 · %19,0

Buffer + DFF + Combinational — tasarımın kendisi

Fiziksel altyapı · 291.635 · %81,0

Fill + Decap + Tap + Diode + Tie — mantık değil, üretilebilirlik

Bellek makroları

16 x

SRAM makrosu

IMEM 4 · DMEM 4 · YZ act_buf 4 · YZ ifm_buf 2 · QSPI FIFO 2

4 x

ROM makrosu

YZ ağırlık ROM'u — u_wrom0..3



Çip Tasarımında Karşılaşılan Zorluklar

Makrolara güç dağıtımı (PDN) — birbirinden bağımsız iki kök neden

Sorun 1 · Makro güç pinleri boşa kaldı

Belirti Makro güç bağlantısı hiç oluşmuyor; her makro için “not connected to any power/ground nets” uyarısı.

Kök neden Wrapper'lar makroyu vccd1/vssd1 ile instantiate ediyor, üst modüller bu pinleri port olarak tanımlamıyor → implicit net ile her scope'ta sürülme yerli bir vccd1 oluşuyor; global_connect bunu conflict sayıp 34 bağlantıyı atlıyor.

Çözüm Özel pdn_cfg.tcl — bağlantılar global ağa zorla kuruluyor.

Sorun 2 · Döndürülmüş makrolara via atılamadı

Belirti Döndürülmüş SRAM'lerde güç halkası ile çekirdek ızgarası arasında hiç via yok.

Kök neden LEF'te güç halkası met4 dikey + met3 yatay barlardan oluşuyor. 90° döndürünce met4 barlar YATAY oluyor ve met5 ızgarasıyla paralel kalıyor → kesişim sayısı sıfır.

Çözüm pdn_cfg.tcl'e met3–met5 kuralı eklendi; dikeye dönen met3 barları met5'i kesiyor.

ölçüldü: döndürülmüş SRAM'de met4×met5 kesişimi = 0 · döndürülmemiş ROM'da bar başına 4

Doğrulama

PDN-0231 · PDN-0232

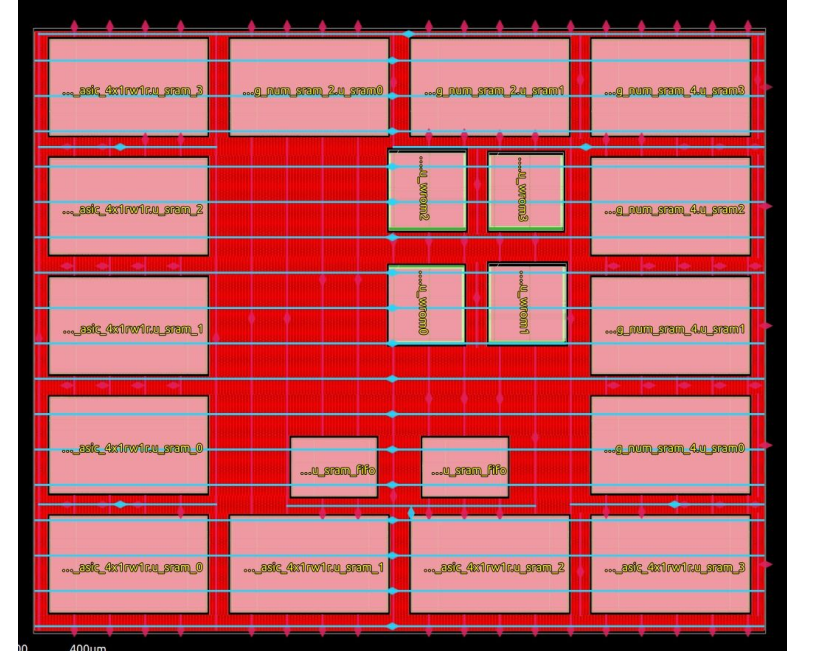
sıfır uyarı

PSM-0040

bütün şekiller bağlı

grid-errors.rpt

boş





Çip Tasarım Akışı: Signoff Karnesi

Fiziksel doğrulama sonuçları — geçen kontroller ve açık kalan işler

Sonuç: DRC, LVS bağlantı, PDN ve STA kontrollerinin tamamı geçti. İki açık madde var: anten diyodu hücresinin PDK SPICE modelinden gelen 1 LVS property uyarısı ve met1 üzerinde 1 anten ihlali — ikisi de işlevselliği etkilemiyor.

Geçen kontroller

Magic DRC	0
Magic illegal overlap	0
Netgen LVS · üst seviye	match · 1 property
STA · 9 PVT köşesi	0 setup / 0 hold
Setup ve Hold TNS	0,0000
Route DRC	0

Açık kalan işler

KLayout DRC	0
ROM makro LVS	0 / 4
Dört ROM'un DRC ve LVS raporları makro dizinlerinde; hepsinde 0 ihlal.	
Antenna ihlali	1 net / 1 pin
1.786 diyot hücresi var (1.159'u anten onarımında eklendi); met1'de tek net 2,02 kat aşıyor (limit 400). Önceki koşumda 0'dı.	



Çip Tasarım Akışı: Teslim Paketi

asic/results/ — üretim ve doğrulama için üretilen çıktıların tamamı

Kapsam: Akış uçtan uca koştu (run/silicore, 13 Eylül 2026) ve zorunlu çıktıların tamamı üretildi. Zamanlama modelleri dokuz PVT köşesi için ayrı ayrı çıkarıldı.

GDSII 3 dosya

Maske verisi, akışın nihai çıktısı

LEF 1 dosya

Abstract view, üst seviye entegrasyon

DEF 1 dosya

Place & route sonucu

Netlist 3 dosya

Sentez · PnR · PDN eklenmiş

SDF 9 köşe

SDF back-annotation (GLS için)

SPEF 3 dosya

Parasitic (RC) extraction

LIB 9 dosya

Timing model, ROM'ları kendi ürettiğimiz

SPICE + MAG 2 dosya

LVS karşılaştırması ve layout kaynağı

SDC 2 dosya

Timing constraint'ler



Çip Tasarım Akışı: Gate-Level Simulation (GLS)

Sentez sonrası netlist boot ediyor ve kopyalanan programı gerçekten koşuyor. Sentezlenmiş netlist üzerinde test edildi.

GLS'de doğrulanan dokuz kontrol

- ✓ QSPI CS düştü, flash işlemi başladı
- ✓ QSPI SCK saati üretildi
- ✓ IMEM dword sayısı = 2048 (beklenen değer)
- ✓ IMEM içeriği firmware.hex ile birebir aynı (942 dword)
- ✓ gpio_out_pins X içermiyor
- ✓ uart0_tx X değil
- ✓ I2C SCL X değil
- ✓ CPU IMEM'den kod koşuyor, UART0'a bayt çıktı
- ✓ UART0'da firmware başlığı "V06" seri port'ta görüldü

Sonuç

Kontrol	9 / 9
Netlist hücre sayısı	67.076

```
source verification/systest.tcl  
systest gls (Vivado Tcl Console)
```



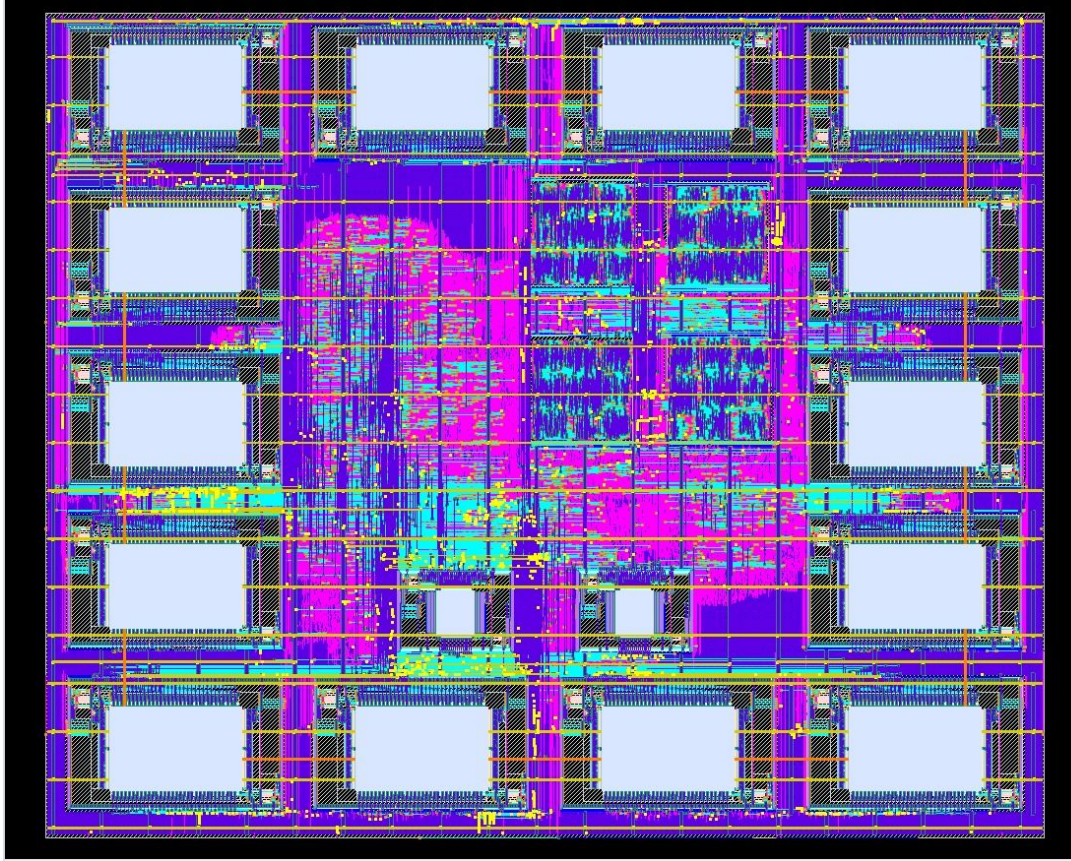
T.C. SANAYİ VE
TEKNOLOJİ BAKANLIĞI

#MİLLİ
TEKNOLOJİ
HAMLESİ



Çip Tasarım Akışı: Nihai GDSII Serimi

Üretime hazır layout — 16 SRAM + 4 ROM makrosu, standart hücre mantığı · sky130A



Öne çıkanlar

Die alanı

8,32 mm²

Die boyutu

3172,4 × 2622,7 μm

SRAM makrosu

16

ROM makrosu

4



Kaynakça

- [1] Texas Instruments, "A Basic Guide to I2C", Application Report SBAA565.
- [2] Micron Technology, 256Mb 3V Multiple I/O Serial NOR Flash Memory Datasheet, 2024.
- [3] IEEE Standard for Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1-2013.
- [4] IEEE Standard for Universal Verification Methodology Language Reference Manual, IEEE Std 1800.2-2020.
- [5] OpenHW Group, CORE-V Formal Verification Plan Template (VerifPlan_Template.xlsx), core-v-verif GitHub reposu.
- [6] Ashok B. Mehta, SystemVerilog Assertions and Functional Coverage: Guide to Language, Methodology and Applications, 2nd Ed.
- [7] OpenLane Documentation — <https://openlane2.readthedocs.io/en/latest/>
- [8] "Do we have to disable the magic drc check when using the sram?", Open-Source Silicon Forum (Erişim: 15 Haz. 2026).
- [9] ARM Limited, AMBA AXI and ACE Protocol Specification, ARM IHI 0022H, 2017.
- [10] RISC-V Software Src, Spike RISC-V ISA Simulator — github.com/riscv-software-src/riscv-isa-sim
- [11] SkyWater Technology & Google, SKY130 Open Source PDK — skywater-pdk.readthedocs.io
- [12] R. Timothy Edwards, Magic VLSI Layout Tool — opencircuitdesign.com/magic/
- [13] KLayout — High Performance Layout Viewer and Editor — klayout.de
- [14] OpenRAM: An Open-Source Memory Compiler — openram.org
- [15] cocotb — Coroutine Based Cosimulation Testbench Environment — docs.cocotb.org
- [16] Gowin Semiconductor, GW2A-18 FPGA Datasheet (Tang Primer 20K).
- [17] TensorFlow Lite for Microcontrollers — ai.google.dev/edge/litert/microcontrollers
- **Açık kaynak IP:** CV32E40P işlemci çekirdeği — OpenHW Group.
- **Proje deposu:** github.com/riscchip/2026_teknofest_Silicore



Kaynakça

- **[18]** R. Herveille, J. Sheahan, I²C Slave Model (WISHBONE rev.B2 uyumlu, sentezlenebilir), OpenCores I2C projesi — opencores.org/projects/i2c
- **[19]** Infineon Technologies (eski Cypress/Spansion), S25FL256L 256 Mb (32 MB) 3.0 V SPI Flash Memory — datasheet ve resmî Verilog simülasyon modeli.
- **[20]** Infineon Technologies, S25FL256S 256 Mb (32 MB) MirrorBit Flash — datasheet ve resmî Verilog simülasyon modeli (READ_ID 90h ve RESET F0h komut referansı).
- **[21]** RISC-V Software Src, riscv-tests — RISC-V Buyruk Kümesi Uygunluk Testleri — github.com/riscv-software-src/riscv-tests
- **[22]** AMD/Xilinx, AXI Protocol Checker LogiCORE IP Product Guide, PG101.
- **[23]** AMD/Xilinx, AXI Verification IP (AXI VIP) LogiCORE IP Product Guide.
- **[24]** cocotb uzantı paketleri: cocotbext-axi, cocotbext-i2c, cocotbext-uart — A. Forencich — github.com/alexforencich
- **[25]** Solomon Systech, SSD1306 128×64 Dot Matrix OLED/PLED Segment/Common Driver Datasheet.



T.C. SANAYİ VE
TEKNOLOJİ BAKANLIĞI



Dinlediğiniz İçin Teşekkürler

Silicore MCU Team