



TEKNOFEST

HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ



TEKNOFEST 2023
ÇİP TASARIM YARIŞMASI
Sayısal İşlemci
Kategorisi

MİLLİ
TEKNOLOJİ
HAMLESİ

KASIRGA - KIZIL

901285

TÜRKİYE
TEKNOLOJİ
TAKIMI



kasırgakızıl

The logo consists of a stylized, swooping graphic in blue and orange, followed by the lowercase text "kasırgakızıl".

TÜBİTAK
BİLGEM

The logo features a circular emblem above the word "BİLGEM", with the word "TÜBİTAK" written vertically to its left.

Takım Üyeleri

Danışman: Prof. Dr. Oğuz Ergin

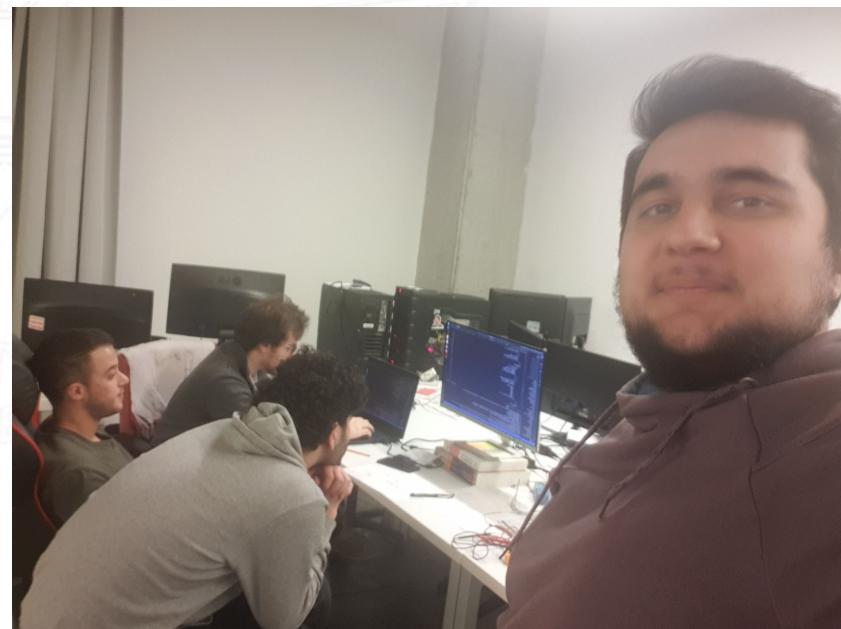
Takım Kaptanı: Seyyid Hikmet Çelik

Üyeler: Emre Hakan Demirli

Mehmet Şakir Aslan

Ahmet Emre Oluk

Ege Deveci



Anahat

- Özeti
- Çip Üstü Sistem
- Çekirdek Tasarımı
- Bellek Tasarımları
 - Bellek Arayüzü
 - Buyruk Ön belleği
 - Veri Ön belleği
- Çevre Birimleri Tasarımları
 - UART
 - SPI
 - PWM
- Çip Tasarım Akışı
- Test ve Doğrulama
- İş Planı
- Kaynakça



Kızıl İşlemci Tasarım Özeti



- İşlemci Getir, Çöz - Yazmaç Oku, Yürüt ve Geri Yaz olmak üzere sıralı 4 boru hattı aşamasından oluşmaktadır.
- Bir seviyeli önbellek yapısı kullanılmıştır. Veri ve buyruk önbelleği eşit (2 KB) boyutlardadır ve toplam önbellek boyutu 4 KB'tır.
- Çipin seriminin gerçekleştirilmesi OpenLane akışı ile sağlanmıştır.
- CSR buyrukları yoktur.

KIZIL İşlemci Tasarım Özeti



Özellikler	KIZIL İşlemci
Buyruk Kümesi	RISC-V (RV32IMCX)
FPGA'ler	AMD Virtex VCU108 (@125 MHz) ✓ Xilinx Basys3 (@60 MHz) ✓ Nexys A7 (@60 MHz) ✓
Veriyolu	Wishbone
Veri Yönlendirmesi	✓
Boru Hattı Aşama Sayısı	4
Veri Önbeliği	2 KB İki Yollu Kümeli İlişkili LRU
Buyruk Önbeliği Boyutu	2 KB Doğrudan Eşlemeli
Çevre Birimi Desteği	UART, SPI, PWM
Nexys A7 Coremark Skoru	153
Coremark / MHz	2.55
Nexys A7 Dhystone Skoru	123227 Dhystone per second, 8 microseconds for one run Dhystone (@60 MHz, 2 milyon iterasyon)
Openlane Serimi Frekans	40 MHz
Openlane Serimi Alan	2500um x 2500um = 6.25 mm ²
Openlane Serimi Güç	124 mW
Openlane Sonrası Kapı Seviyesi Netlist Simülasyon	riscv testlerini geçiyor ✓

İsterlerin Sağlanması

Sayısal Tasarım Kategorisinde Başarı Kriterleri	KIZIL İşlemci
Tasarımların github reposuna yüklenerek düzenli periyotlarla güncellenmesi	✓
İşlemcinin şartnamedeki tüm 32 bit talimat uzantılarını desteklemesi	✓
Modüllerin SoC seviyesinde birleştirilmiş ve çalışır durumda olması	✓
SoC'un test ve benchmark yazılımlarını başarı ile tamamlaması	✓
Tasarım çıktıları bölümünde verilen çıktıların top modül için elde edilmesi	✓
SoC'un serim sonrası testleri başarıyla geçmesi	✓

Özgün ve Özgün Olmayan Tasarımlar



- Çarp Biriktir birimi
 - <https://github.com/temelmertcan/multgen>
- PWM alt modülleri
 - https://github.com/SerdarUnal132/pwm_openmpw
- Toplama birimi (sklansky)
 - https://github.com/tdene/synth_optadders
- **Geriye kalan modüllerin hepsi özgün ya da esinlenilerek oluşturulmuş tasarımlardır. (Kapı Seviyesi Önbellekler Bile!!!)**

Performans Artışı İçin Yapılan İşlemler



- Veri yönlendirmesi
- Flip-flop tabanlı el yapımı önbellekler
- Buyruk önbelleği hizasız erişim yöntemi
- Gshare dallanma tahmini
- İki yolu veri önbelleği
- Vivado Post-route Physical Optimization
- Frekansı artırmak ve çevrim sayısını azaltmaya yönelik diğer optimizasyonlar

Zorlanılan Kısımlar



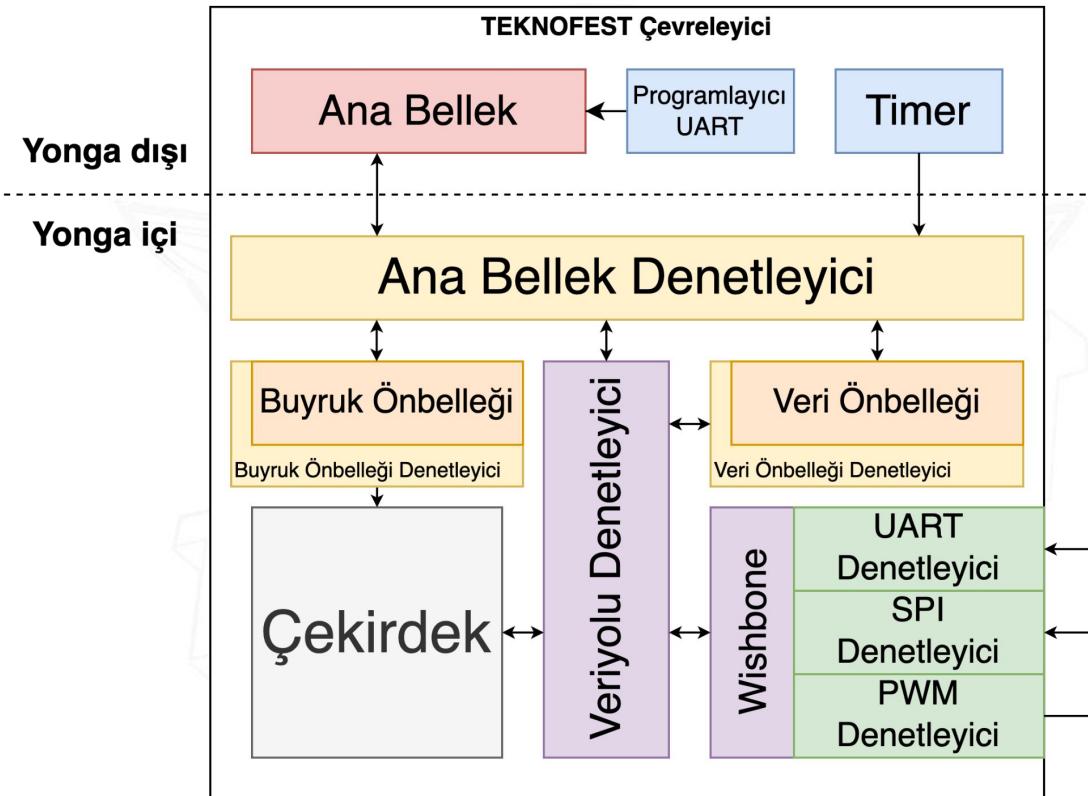
- Openlane → Deneye yanıyla çözdük.
- Debug → Rastgele testler ile çözdük.
- Optimizasyon → Olduğu kadar, sınırlarımıza ulaşmaya başladık.
- Raporlama → Anlatım, işlemciden daha önemli belki de.

DTR'den Bu Yana Değişiklikler



- Önbellekler için; DFFRAM ve OpenRAM yerine, asenkron erişilebilen flip-flop tabanlı el yapımı kapı seviyesi RAM'ler kullanıldı.
- BTB silindi, onun yerine toplayıcı tabanlı program sayacı hesaplayıcı konuldu.
- Toplama işlemi sklansky algoritmasına çevrildi.
- Bellekler denetleyicilerin içinden çıkarılarak en üst modül işlemci (user_processor) modülüne taşındı.
- Çekirdek daha modüler hale getirildi.
- Performans artışına yönelik diğer optimizasyonlar yapıldı.

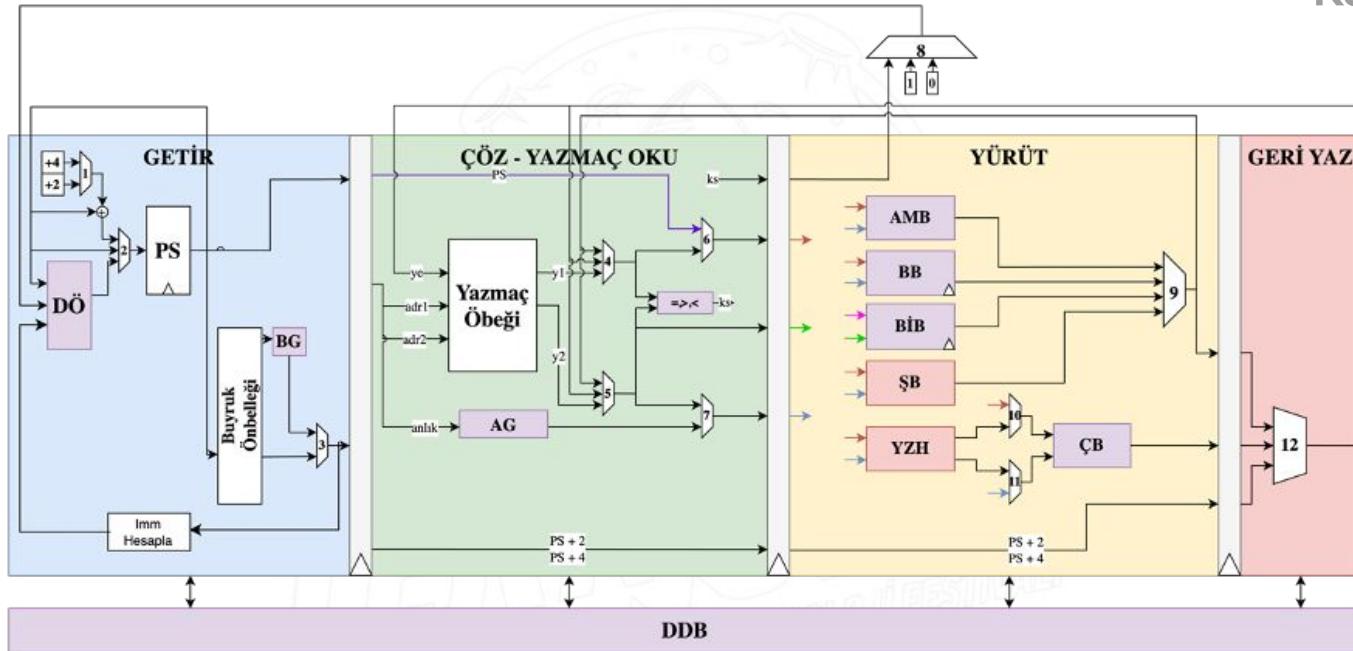
Çip Üstü Sistem



Çekirdek Tasarımı

- RV32IMCX özelleştirilmiş buyruk kümesi mimarisini desteklemektedir.
- Çekirdek, 4 aşamalı sıralı yürütme yapmaktadır. (Getir, Çöz-Yazmaç Oku, Yürüt, Geri Yaz)
- 4 aşamanın sebebi basit ve hızlı bir işlemci tasarlamak ve başarımı artttırmak için boş çevrimler geçirmemek.

Çekirdek Tasarımı - Boru Hattı



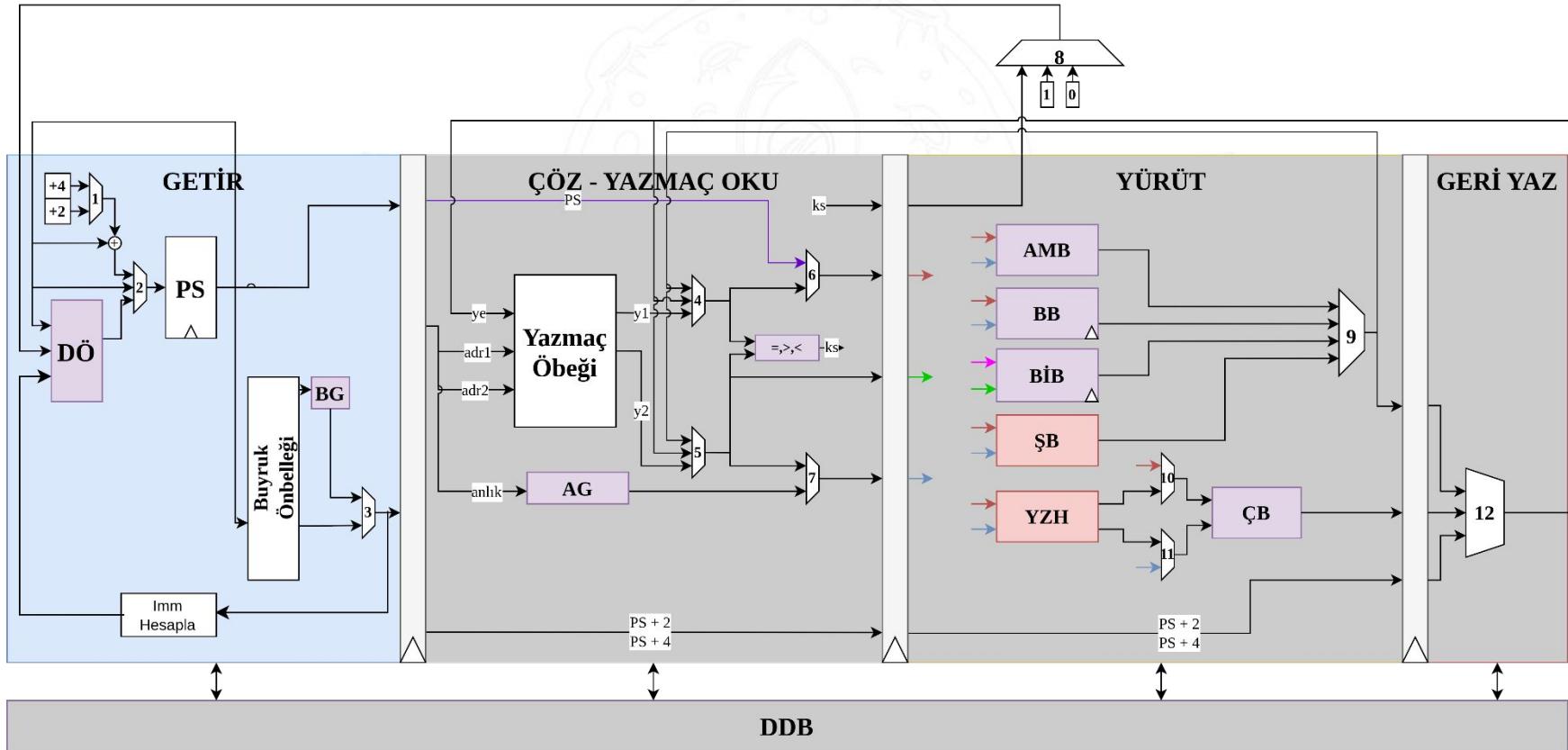
PS: Program Sayacı
=>,<: Aritmetik Karşılaştırıcı
BB: Bölme Birimi
ŞB: Şifreleme Birimi
ÇB: Çarpma Birimi
BG: Büyruk Genişletici
ye: Yaz Etkin
y1: Yazmaç 1 Değeri
adr1: Yazmaç 1 Adresi

DÖ: Dallanma Öngörüci
AMB: Aritmetik Mantık Birimi
BİB: Bellek İşlem Birimi
YZH: Yapay Zeka Hızlandırıcı
DDB: Denetim Durum Birimi
AG: Anlık Genişletici
ks: Karşılaştırma Sonucu
y2: Yazmaç 2 Değeri
adr2: Yazmaç 2 Adresi

Çoklayıcılar:

- 1: Program Sayacı Çoklayıcısı
- 2: Atlama Çoklayıcısı
- 3: Sıkıştırılmış Büyruk Çoklayıcısı
- 4: Yazmaç 1 Çoklayıcısı
- 5: Yazmaç 2 Çoklayıcısı
- 6: Kaynak 1 Çoklayıcısı
- 7: Anlık Değer Çoklayıcısı
- 8: Dallanma Sonucu Çoklayıcısı
- 9: Yürüt Sonuç Yönlendirme Çoklayıcısı
- 10: Çarpma Kaynak 1 Çoklayıcısı
- 11: Çarpma Kaynak 2 Çoklayıcısı
- 12: Geri Yaz Çoklayıcısı

Çekirdek Tasarımı - Getir



Çekirdek Tasarımı - Getir

- Getir, program sayacının belirlenmesi, buyrukların bellekten çekirdeğe getirilmesi ve dallanma öngörülerinin yönetilmesi görevlerinden sorumludur.

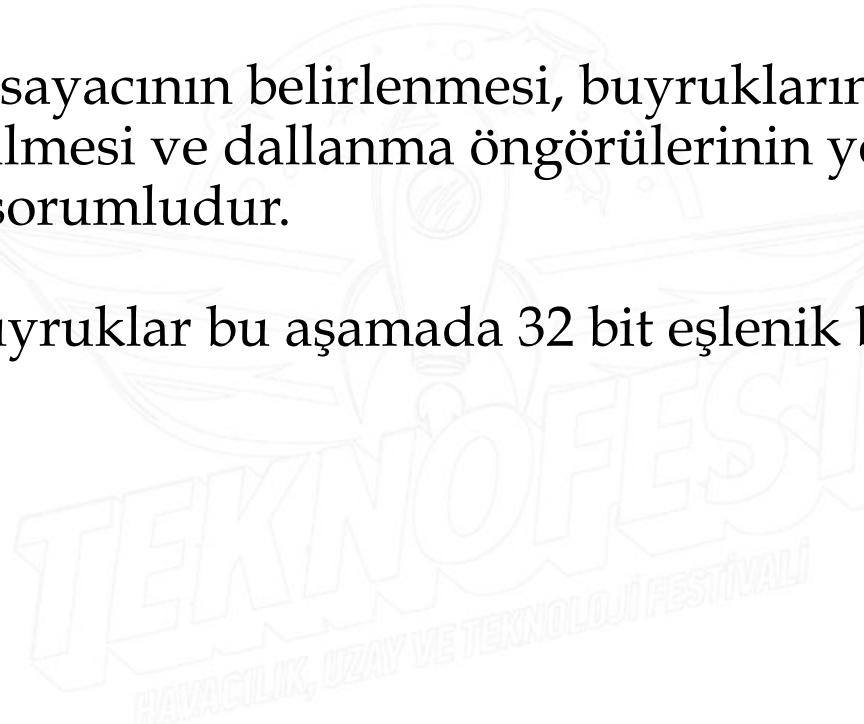
0x40080000 → 'b1000_0000_0000_0000_0000

Adres Alanının alabileceği maksimum değere göre program sayacı kısaltılmıştır.
16 bit aligned olduğu için son bit yok.

Çekirdek Tasarımı - Getir

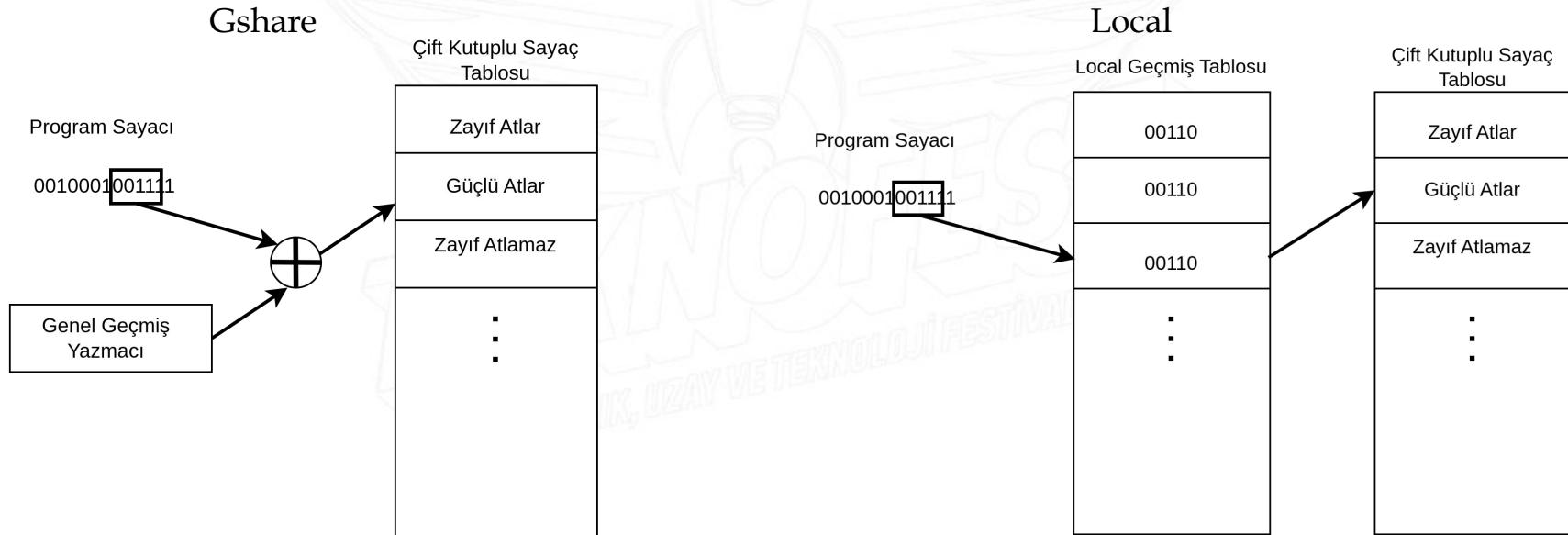


- Getir, program sayacının belirlenmesi, buyrukların bellekten çekirdeğe getirilmesi ve dallanma öngörülerinin yönetilmesi görevlerinden sorumludur.
- Compressed buyruklar bu aşamada 32 bit eşlenik buyruklara çevrilir.



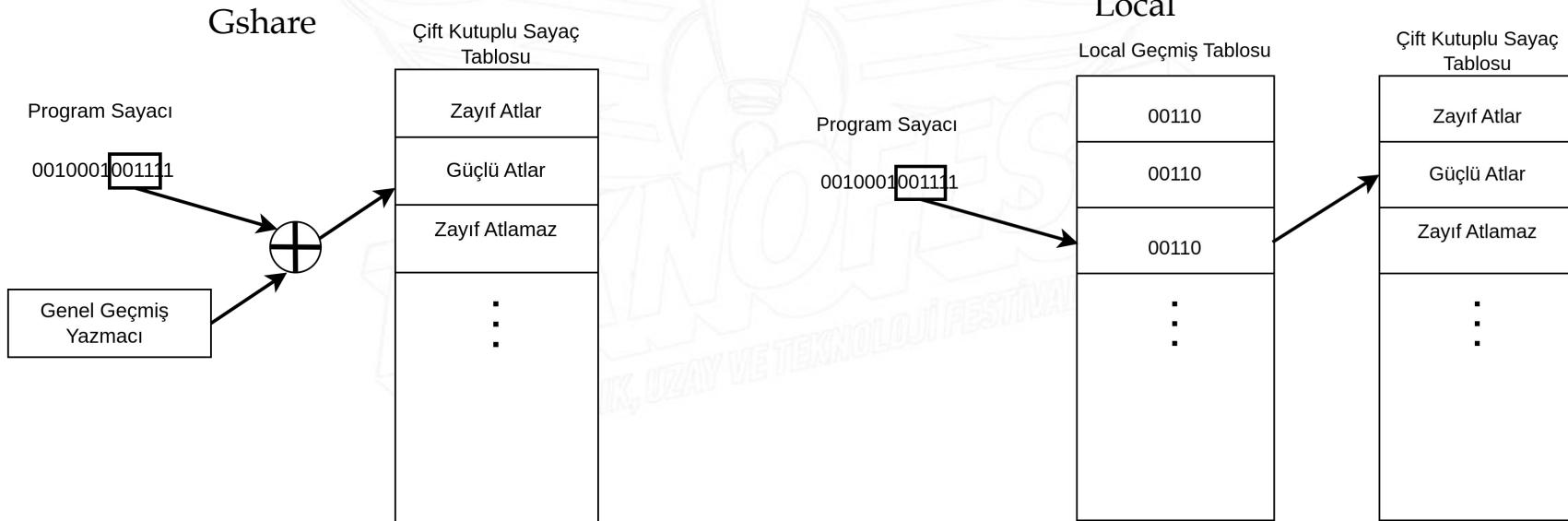
Çekirdek Tasarımı - Dallanma Öngörücü

- İşlemcide kullanılmak üzere Gshare ve Local Öngörüçüler gerçekleştirilemiştir.



Çekirdek Tasarımı - Dallanma Öngörücü

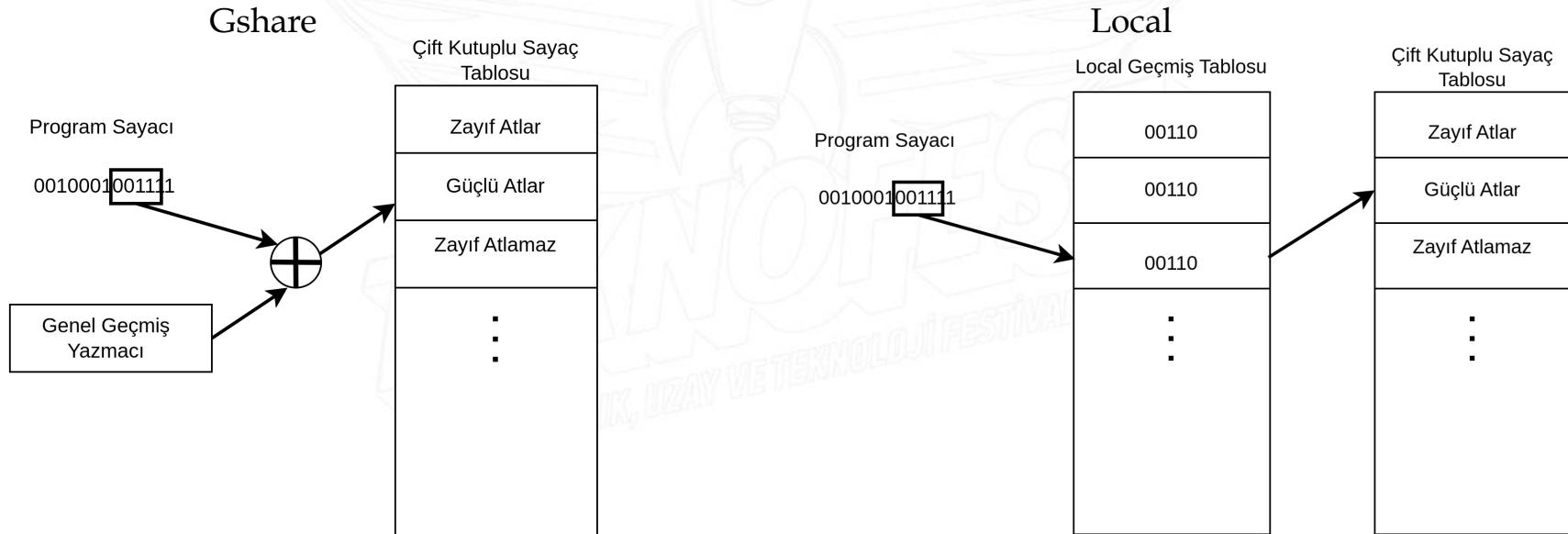
- GShare öngörücü genel geçmiş ve yerel geçmiş verilerini xorlayarak indeks hesaplarken local öngörücü her satır için yerel bir geçmiş tutar ve her iki öngörücü de 32 satırlı (5 bit) 2 bitlik sayaç tablosunu indeksler.



Çekirdek Tasarımı - Dallanma Öngörücü

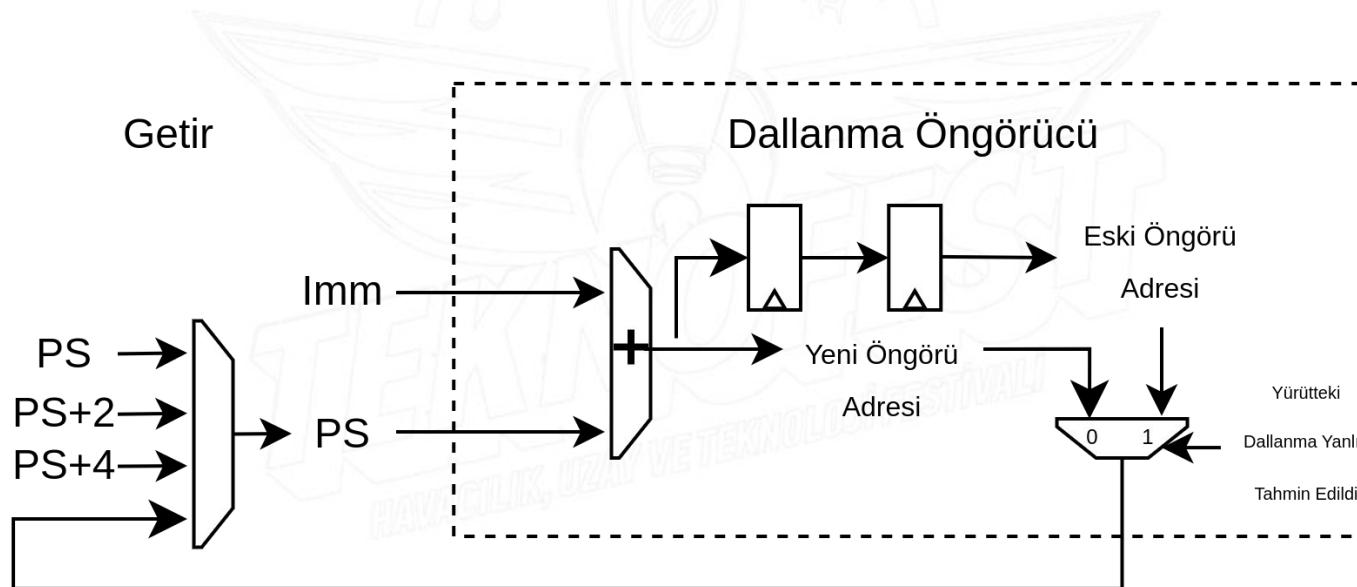


- Daha yüksek performansa ve daha az alana sahip olması nedeniyle ana işlemcide Gshare kullanmasına karar verilmiştir.



Çekirdek Tasarımı - Dallanma Öngörücü

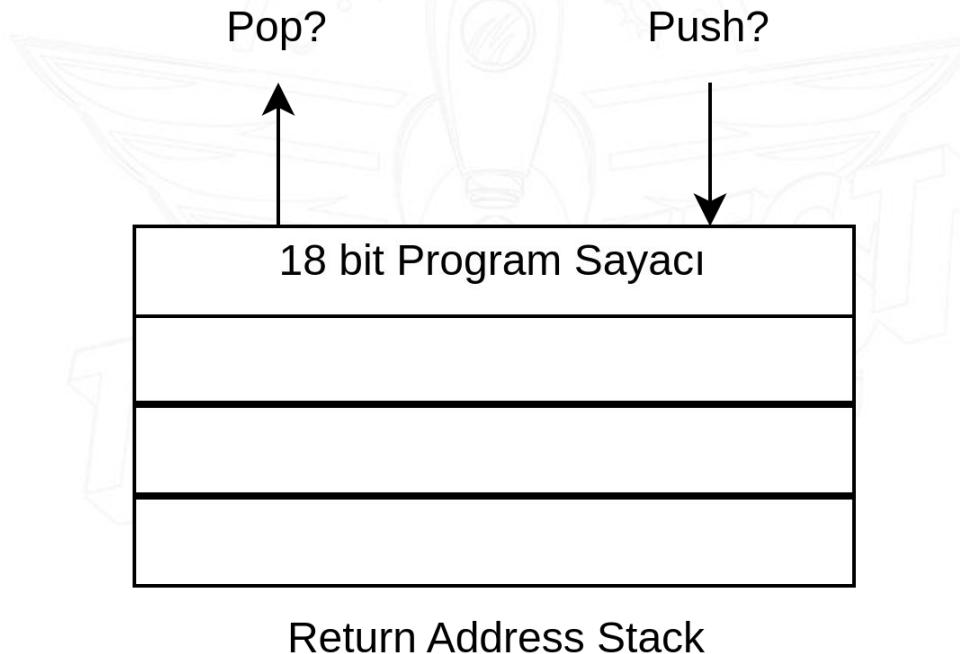
- İşlemcide atlama adresi getirde hesaplanmakta ve dallanma buyruğu işlemciden çıkışa kadar içерideki yazmaçlarda tutulmaktadır.



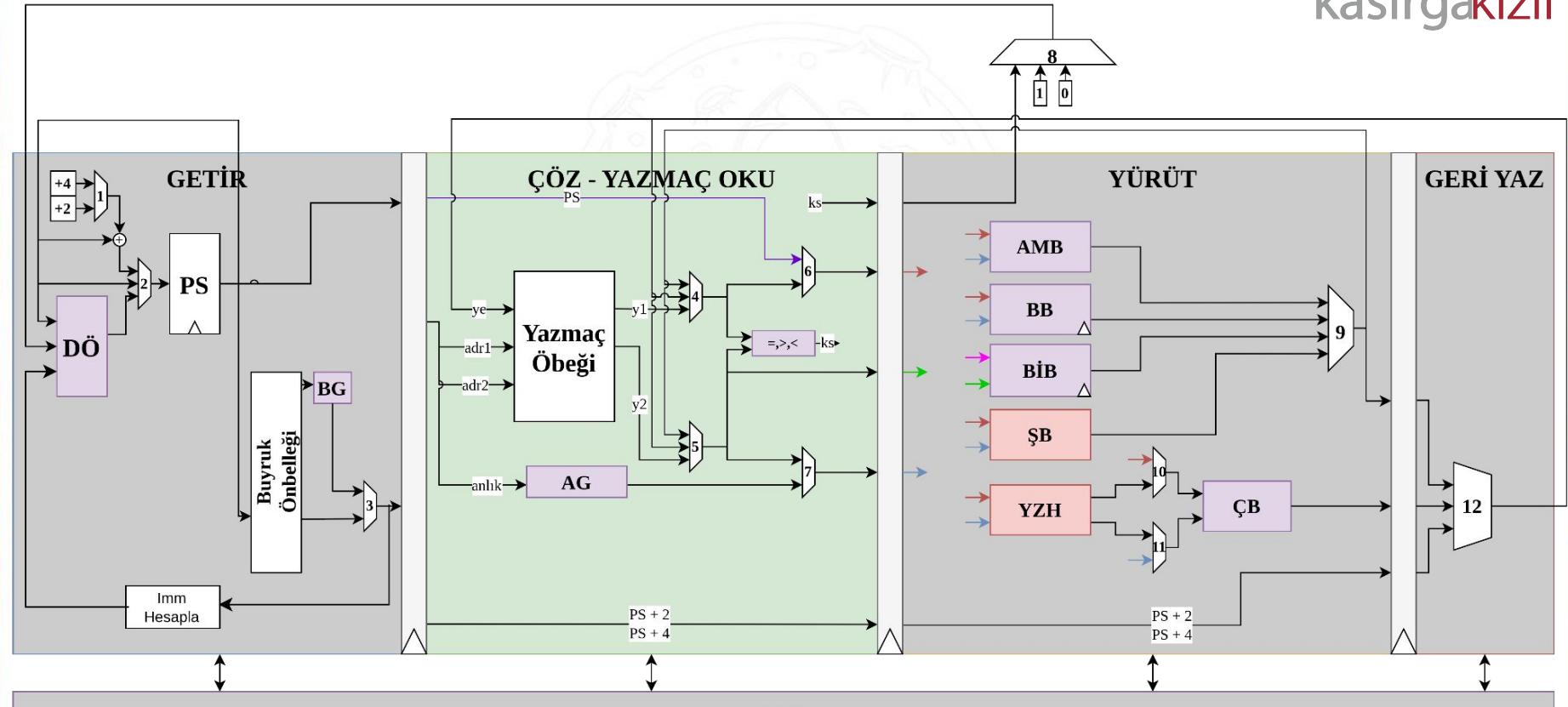
Çekirdek Tasarımı - Dallanma Öngörücü



- Fonksiyon çağrıları için 4 satırı sahip bir RAS bulunmaktadır.



Çekirdek Tasarımı - Çöz Yazmaç Oku



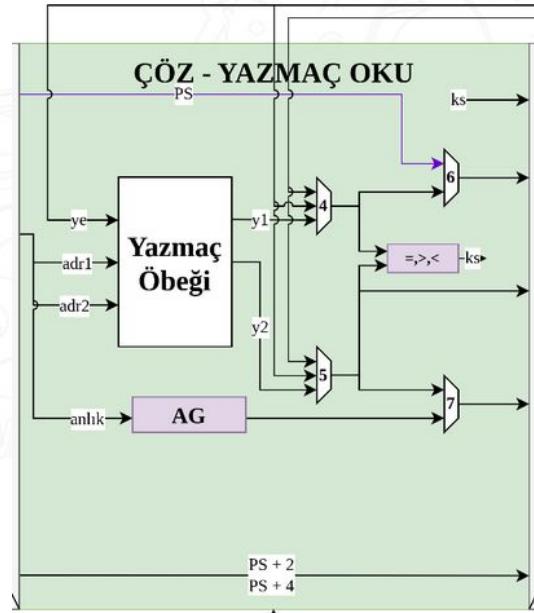
Çekirdek Tasarımı - Çöz Yazmaç Oku



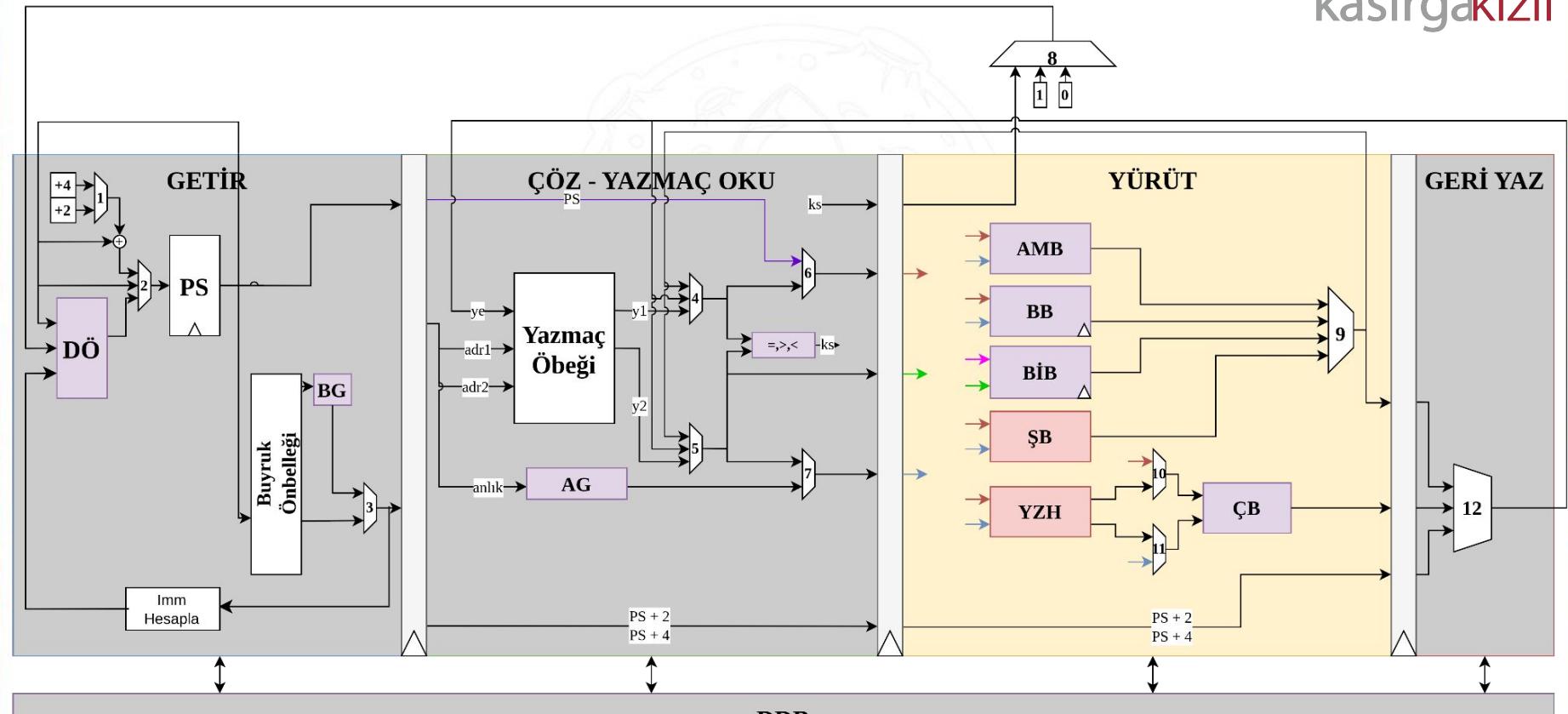
- Getirilen buyruk ait olduğu mikro işleme çevrilir.
- Veri bağımlılığı yönlendirme ile çözülür.
- Dallanma buyruklarının karşılaştırması yapılır.
- Anlık genişletilir.

```
`define ADD_MI
`YZH_YOK, `SIFRELEME_YOK, `BIB_YOK, `CARPMA_YOK,
`BOLME_YOK, `AMB_TOPLAMA, `DAL_YOK, `BIRIM_AMB,
`OPERAND_REG, `YAZMAC_YAZ, `GERIYAZ_KAYNAK_YURUT}
```

```
always @* begin
    // Cozulmesi gereken bitler 14 bit 30:29, 27, 25, 21:20, 14:12, 6:2
    // bitleri en tamam olanдан olmayana kadar gitmek gerek.
    casez(buyruk_coz_w)
        `EBREAK_COZ: begin mikroiislem_sonraki_r = `NOP_MI; end
        `ECALL_COZ: begin mikroiislem_sonraki_r = `NOP_MI; end
        `CONV_CLR_W_COZ: begin mikroiislem_sonraki_r = `CONV_CLR_W_MI; end
        `CONV_CLR_X_COZ: begin mikroiislem_sonraki_r = `CONV_CLR_X_MI; end
```

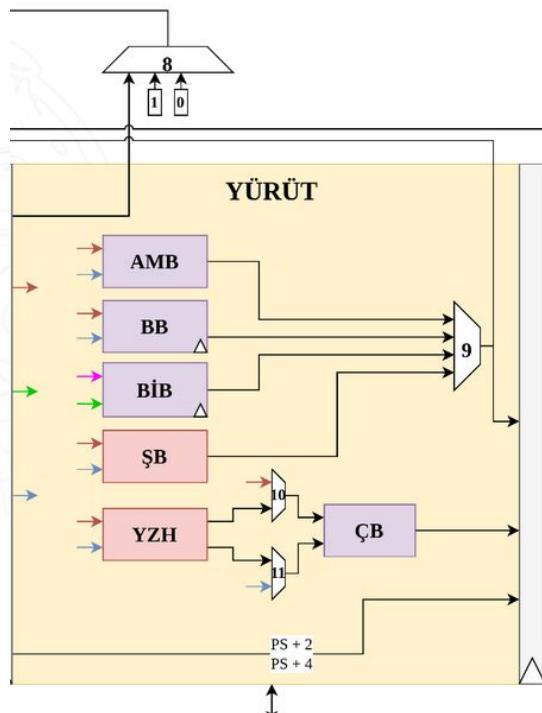


Çekirdek Tasarımı - Yürüt

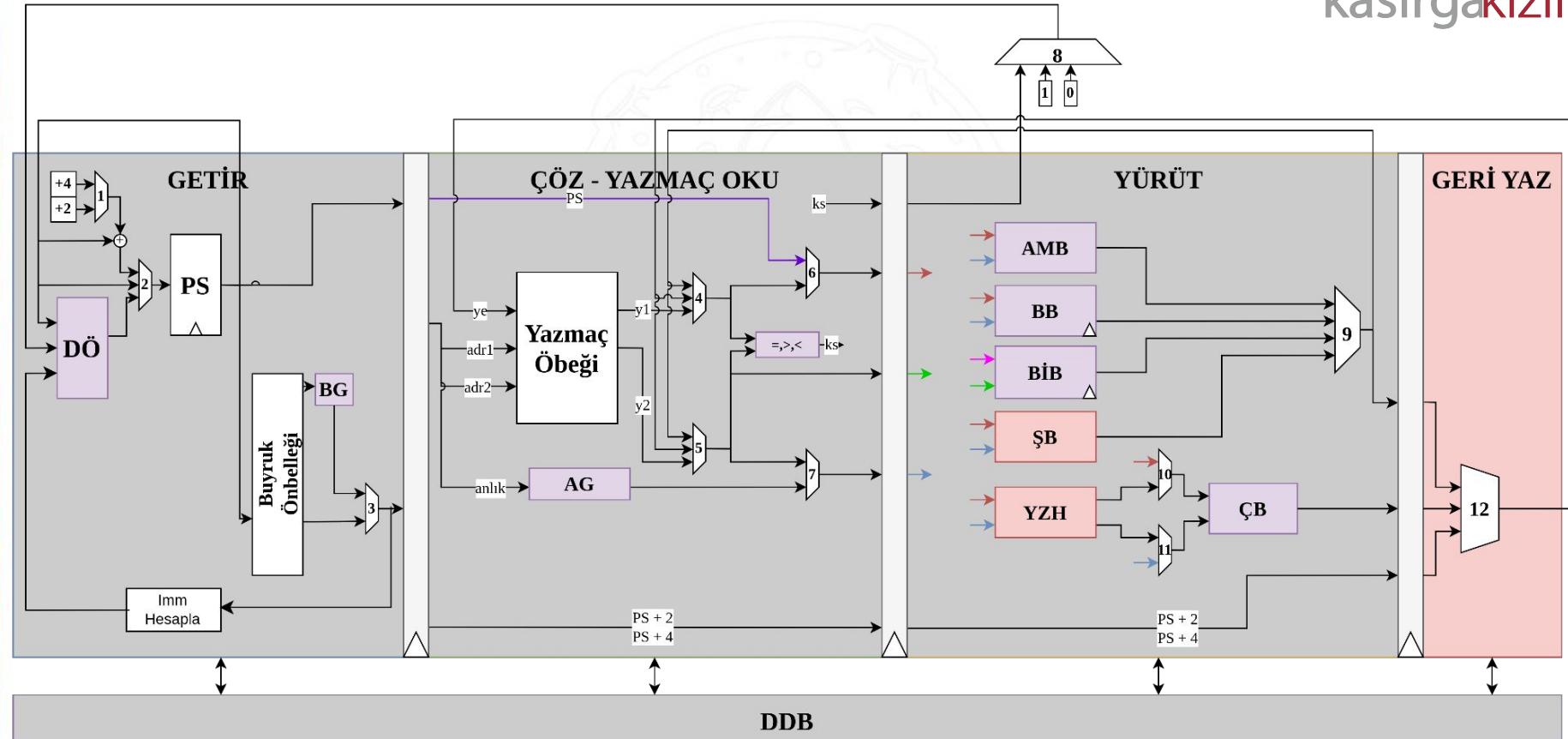


Çekirdek Tasarımı - Yürüt

- AMB:
 - Sklansky Toplayıcı
 - PC+Imm
- BİB:
 - Byte ve Half Word işlemleri için veri kaydırır
- SB:
 - Şifreleme buyrukları
- YZH:
 - Yazmaçlar ve kontrol sinyalleri
- ÇB:
 - Modified booth dadda çarpanı ve biriktirici
- BB:
 - 19 çevrim kaydetmesiz bölme
- Dallanma ve atlama işlemleri için ayrı bir birim yok.



Çekirdek Tasarımı - Geri Yaz

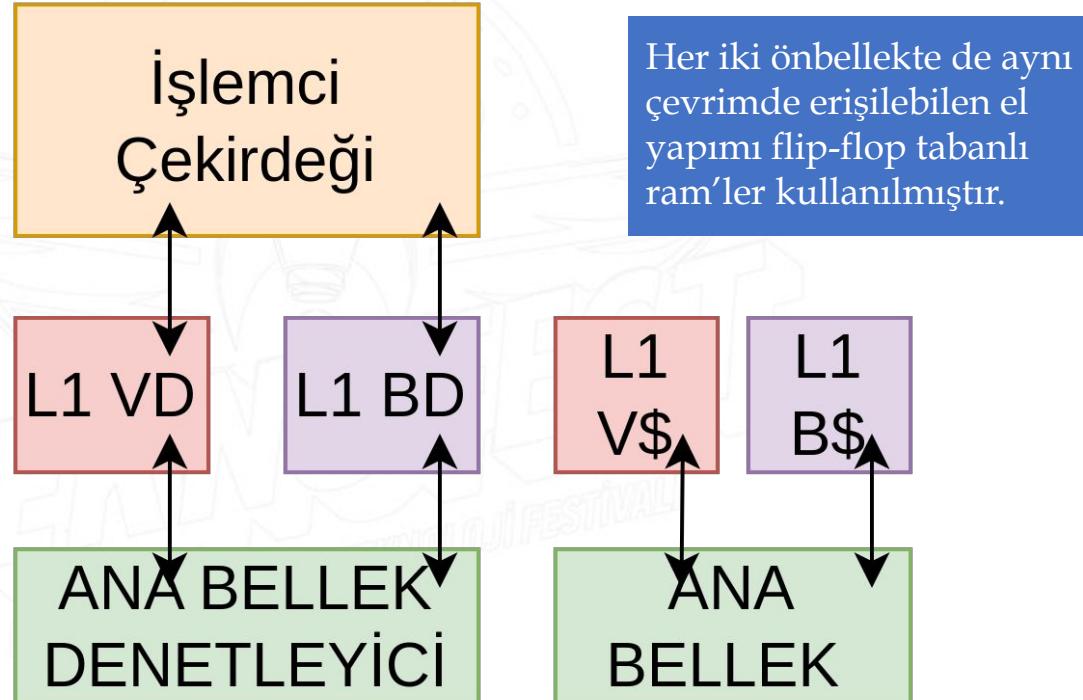


Bellek Arayüzü

Veri Önbellesi 2 yollu kümeli ilişkili olarak tasarlanmıştır, 2KB boyuta ve 256 satıra sahiptir.

Buyruk Önbellesi doğrudan eşlemeli olarak tasarlanmıştır, 2KB boyuta ve 512 satıra sahiptir.

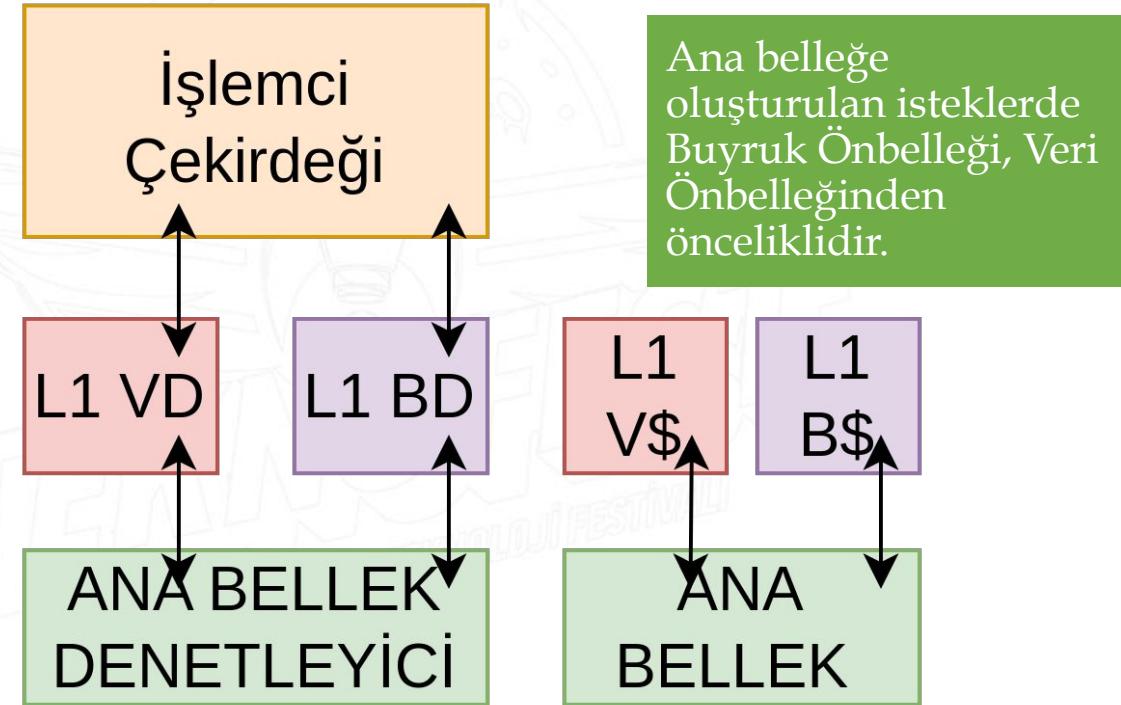
Her iki önbellek de hit Okuma-Yazma 0 çevrimidir.



Her iki önbellek de aynı çevrimde erişilebilen el yapımı flip-flop tabanlı ram'ler kullanılmıştır.

Bellek Arayüzü

Veri ve buyruk önbelleklerinin ana bellekle olan iletişimini ana bellek denetleyicisi tarafından sağlanır.

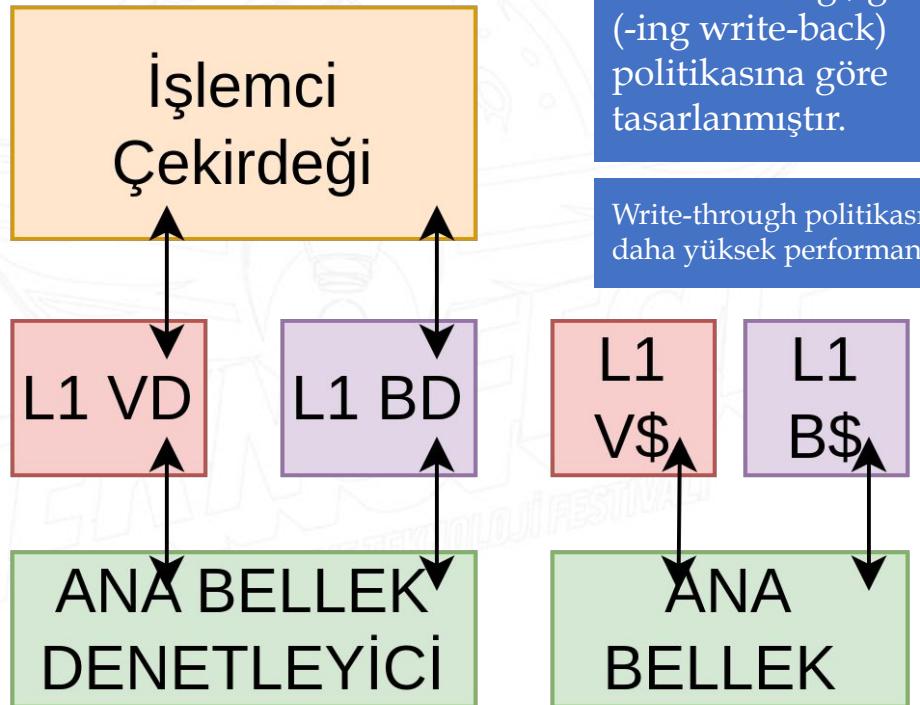


Ana belleğe oluşturulan isteklerde Buyruk Önbelleği, Veri Önbelleğinden önceliklidir.

Bellek Arayüzü

Veri Önbelleğinde yaz ve yerini ayır (-ing. write allocate) politikası kullanılmıştır.

Yazılan verilerin sık sık okunduğu sistemlerde yaz ve yerini ayırma (-ing write-no-allocate)'e göre daha yüksek performansa sahiptir.

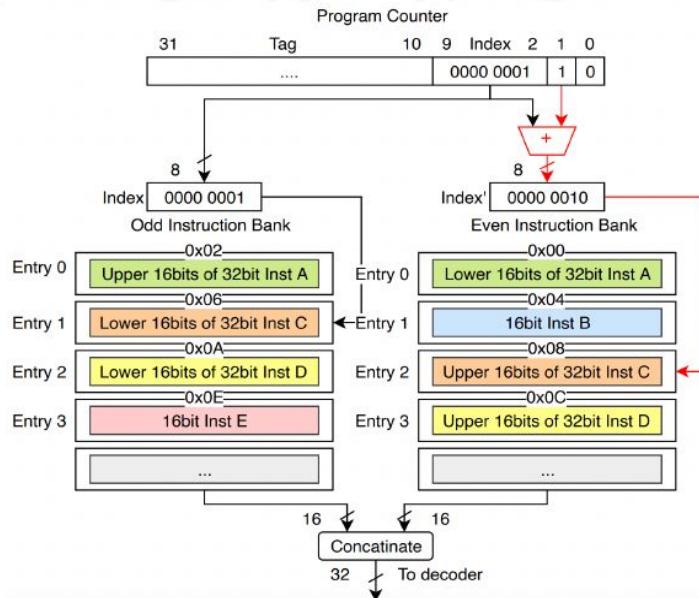


Veri Önbelleği, geri yaz (-ing write-back) politikasına göre tasarlanmıştır.

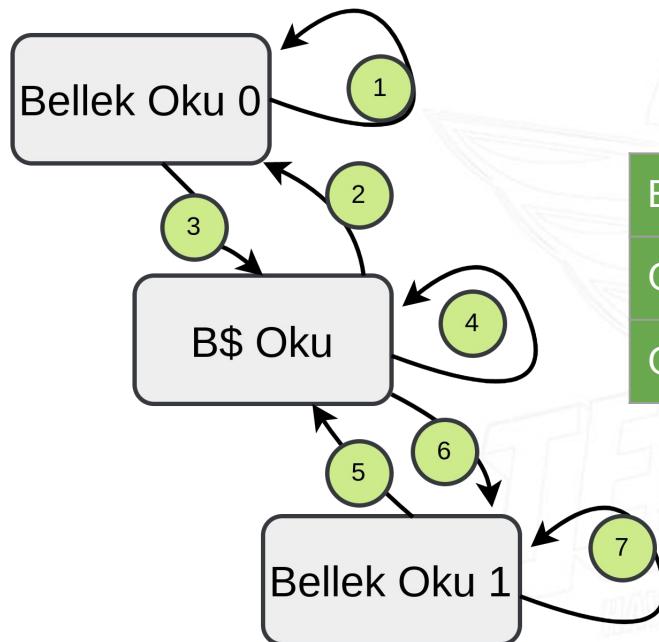
Write-through politikasına göre daha yüksek performansa sahiptir.

Buyruk Önbelleği

- Buyruk önbelleği 16 bitlik adresleme yapar ve her okumada 2 adet 16 bitlik veriyi okur, bu sayede compressed buyruklar için hizasız (unaligned) buyruk belleği erişimleri desteklenmiş olur.
- 8 bit etiket ve 32 bit veri öbeğine sahip



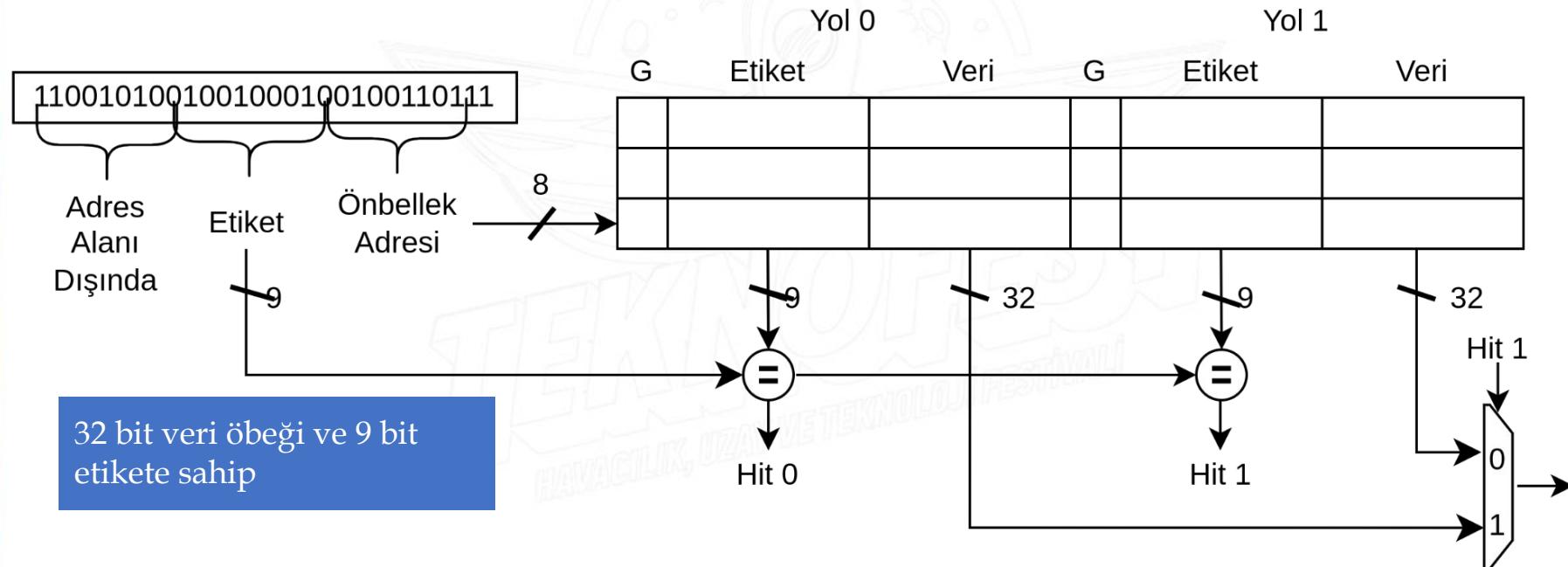
Buyruk Önbellegi - Durum Makinesi



Buyruk Belleği İsteği	Çevrim Sayısı	Durum Makinesi İzlenilen Yol
Okuma (En İyi)	0	4
Okuma (En Kötü)	36	2 -> 1 -> 3 -> 6 -> 7 -> 5

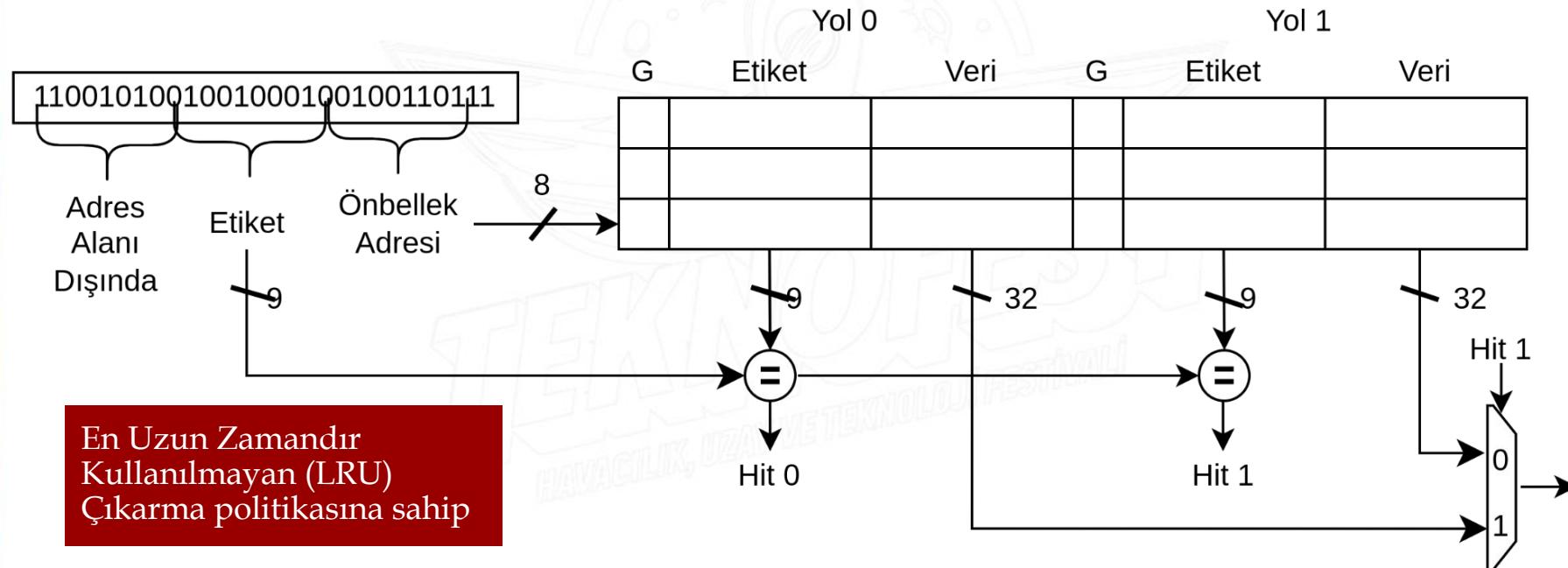
Veri Önbelleği

Veri önbelleği hit olduğu durumda okuma işlemi örneği

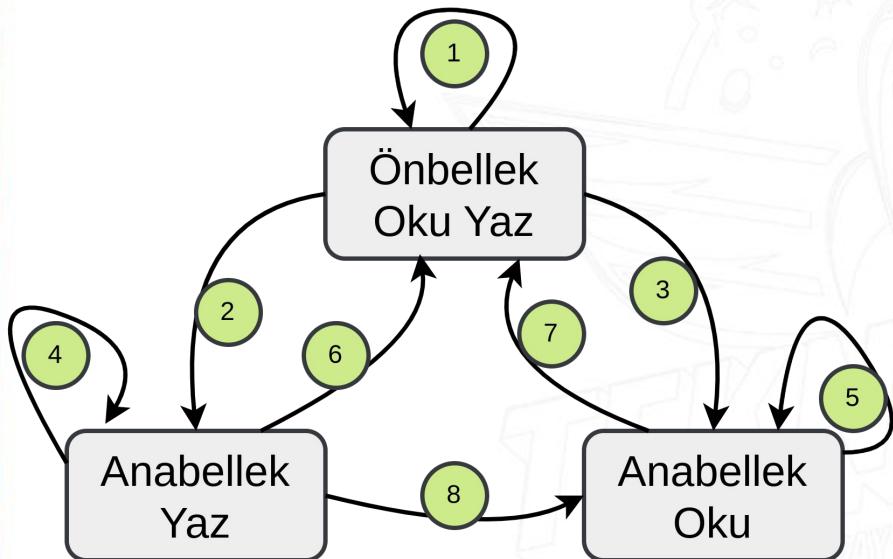


Veri Önbellegi

Veri önbellegi hit olduğu durumda okuma işlemi örneği

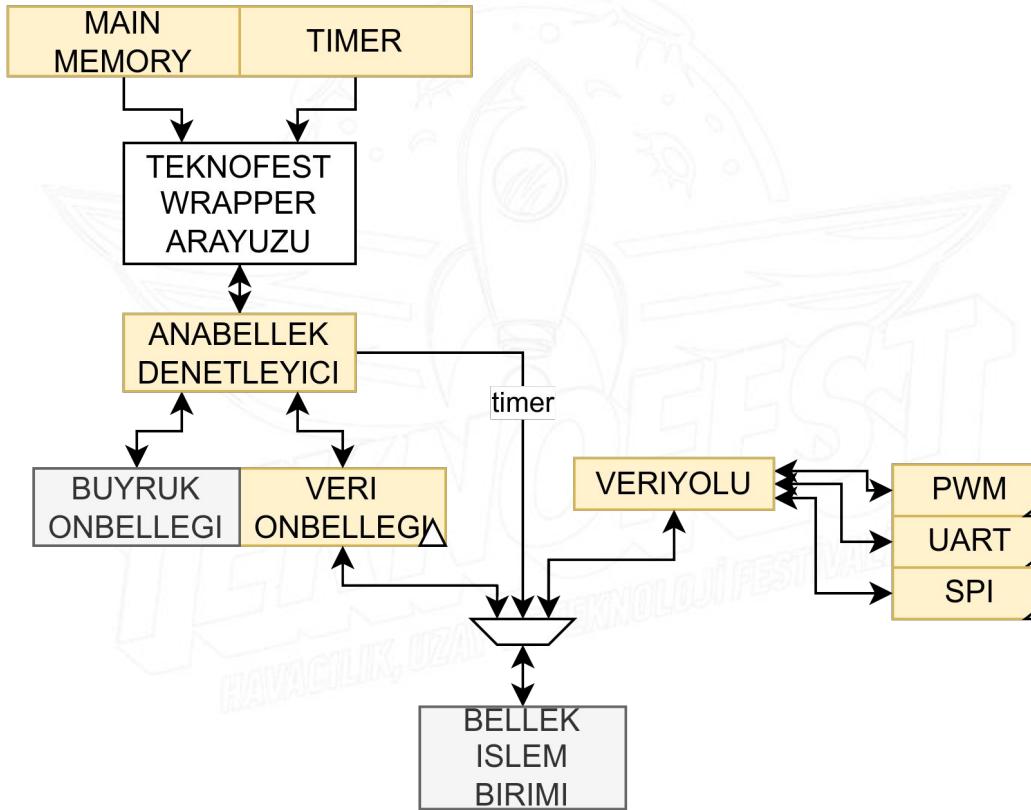


Veri Önbelleği - Durum Makinesi



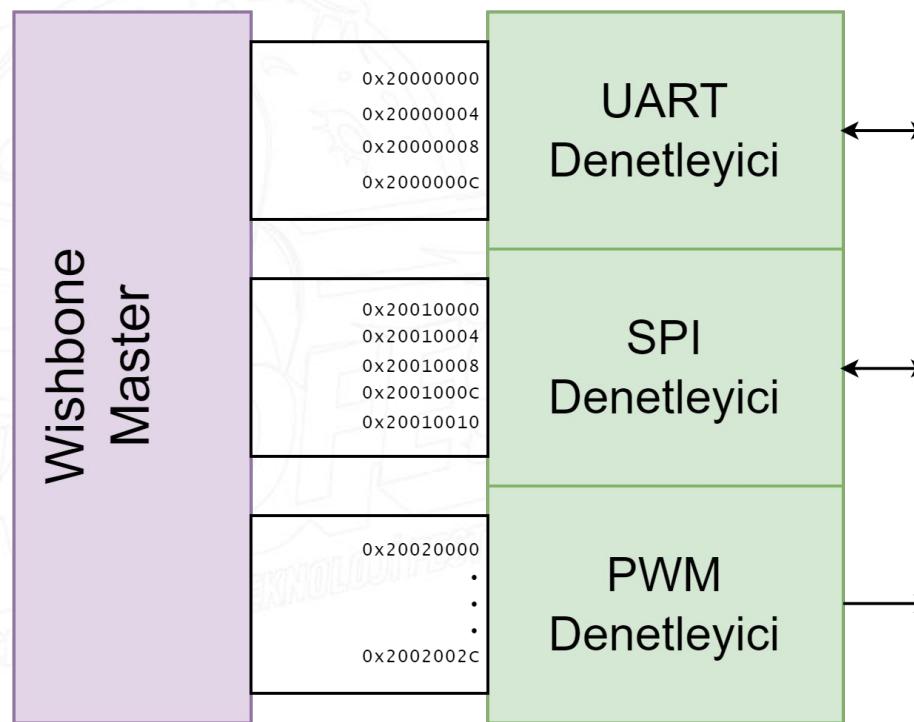
Veri Belleği İsteği	Çevrim Sayısı	Durum Makinesi İzlenen Yol
Yazma (En İyi)	0	1
Yazma (En Kötü)	18	1 -> 2 -> 6
Okuma (En İyi)	0	1
Okuma (En Kötü)	34	1 -> 2 -> 8 -> 7

Sistem ve Çevre Birimleri Üst Yapısı



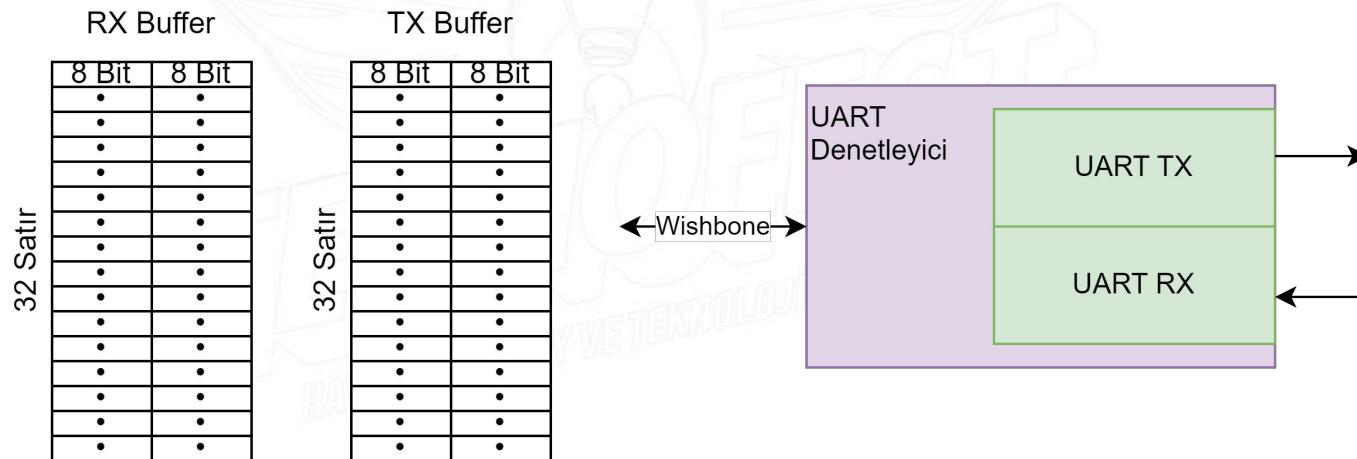
Çevre Birimleri - Wishbone Arayüzü

- Esnek Bağlantı Arayüzü
- 0x20030000 adresine ekstra VGA bağlanıp da denendi.
[\(https://github.com/
KASIRGA-KIZIL/teknoloji-kizil/tree/vgali\)](https://github.com/KASIRGA-KIZIL/teknoloji-kizil/tree/vgali)



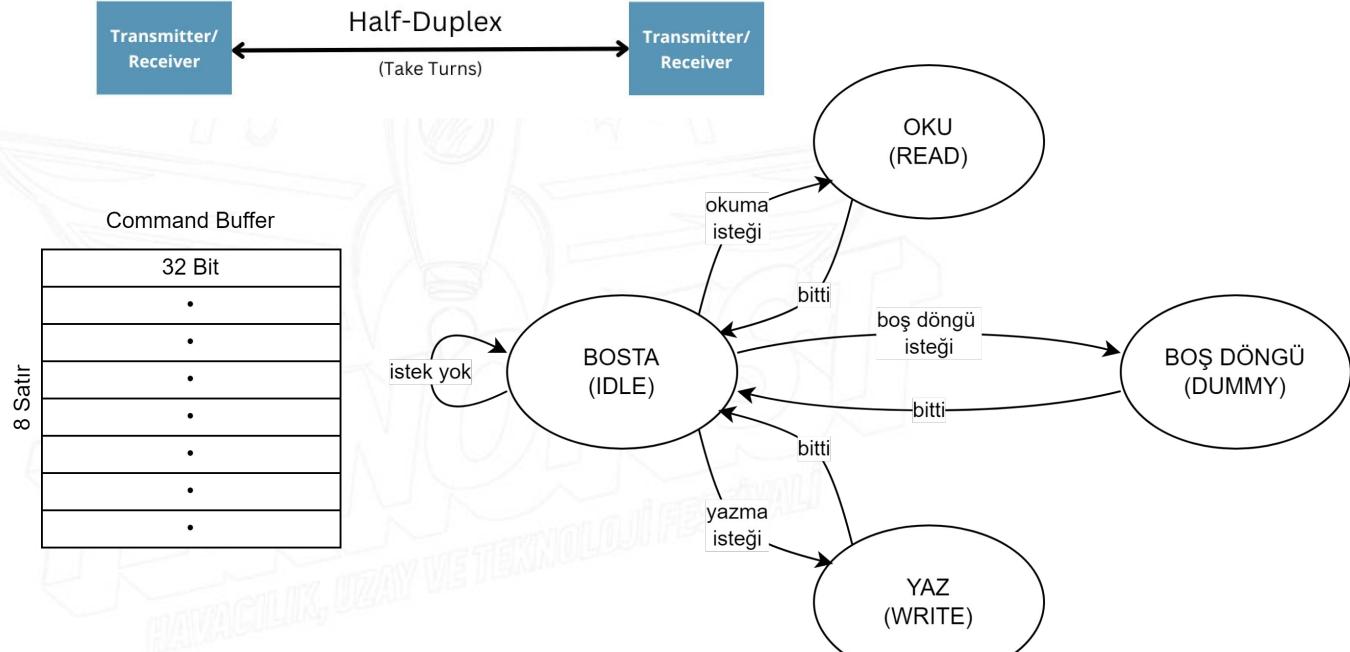
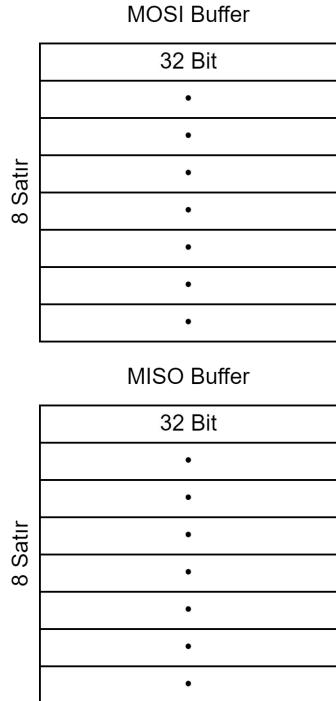
Çevre Birimleri - UART

- FIFO (İlk Giren İlk Çıkar) tabanlı TX ve RX modülleri
- Bunların üzerine yazılan ve şartnamedeki adres alanları ve yazmaçları kapsayan denetleyici modülü bulunmakta



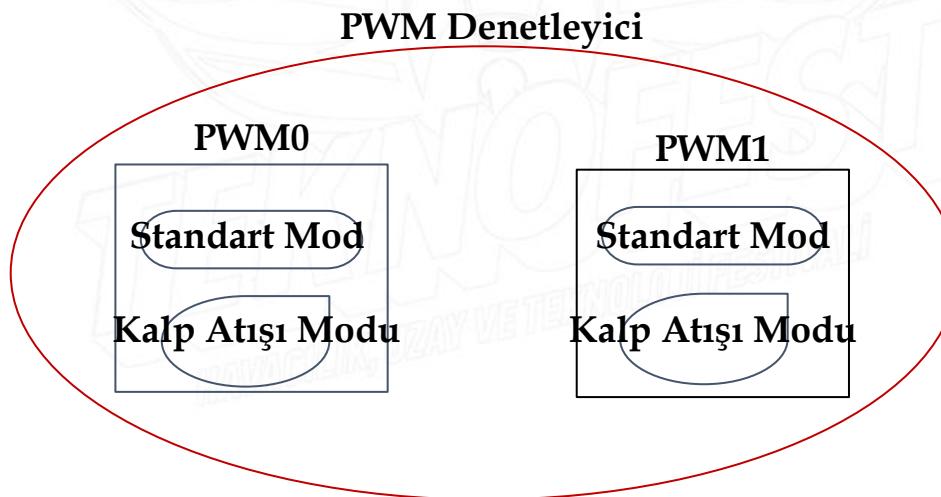
Çevre Birimleri - SPI

- Half - Duplex SPI

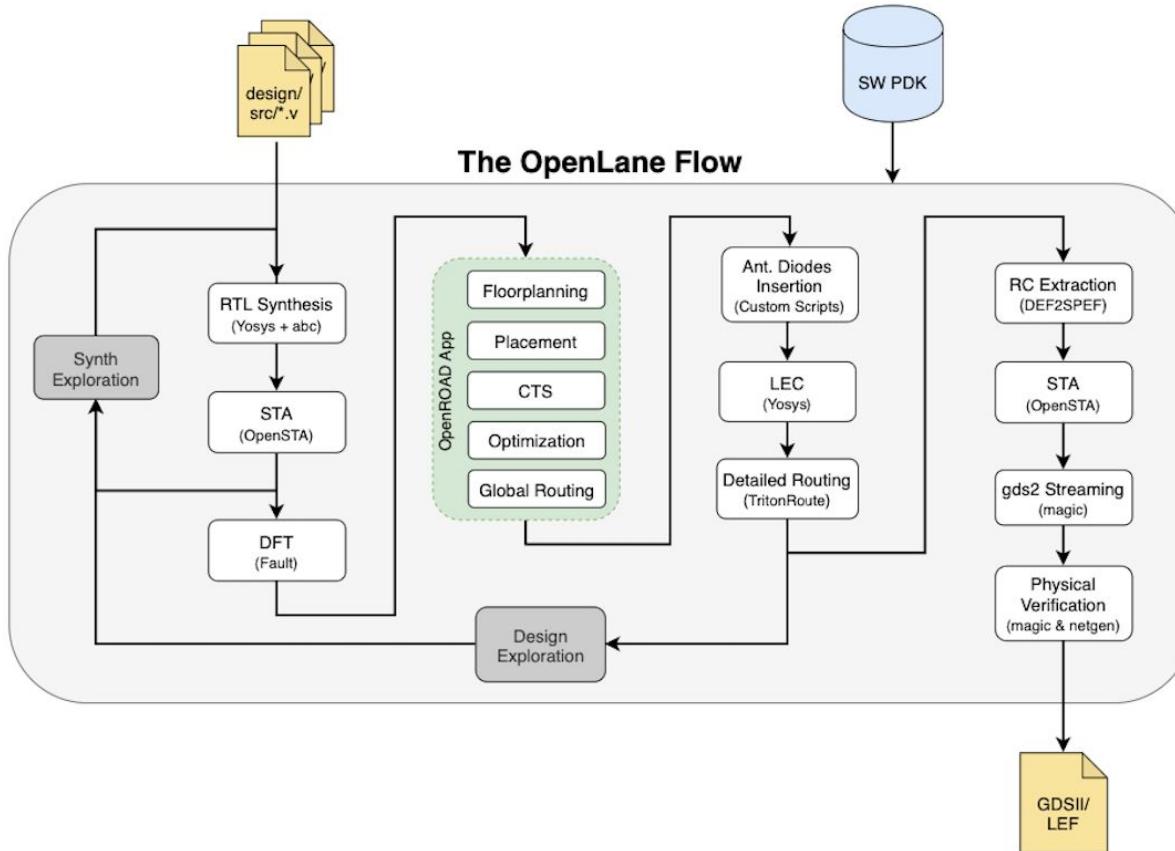


Çevre Birimleri - PWM

- https://github.com/SerdarUnal132/pwm_openmpw reposundan PWM Standart ve Kalp Atışı modlarını sağlayan modüller alınmış, küçük değişiklikler yapılmış ve üzerlerine şartnameyi sağlayan ve yazmaçları bulunduran denetleyici modülü yazılmıştır.



Çip Tasarım Akışı



Çip Tasarım Akışı - Nihai Tasarım

Frekans: **40 MHz**

Alan: **2500um x 2500um (6.25 mm²)**

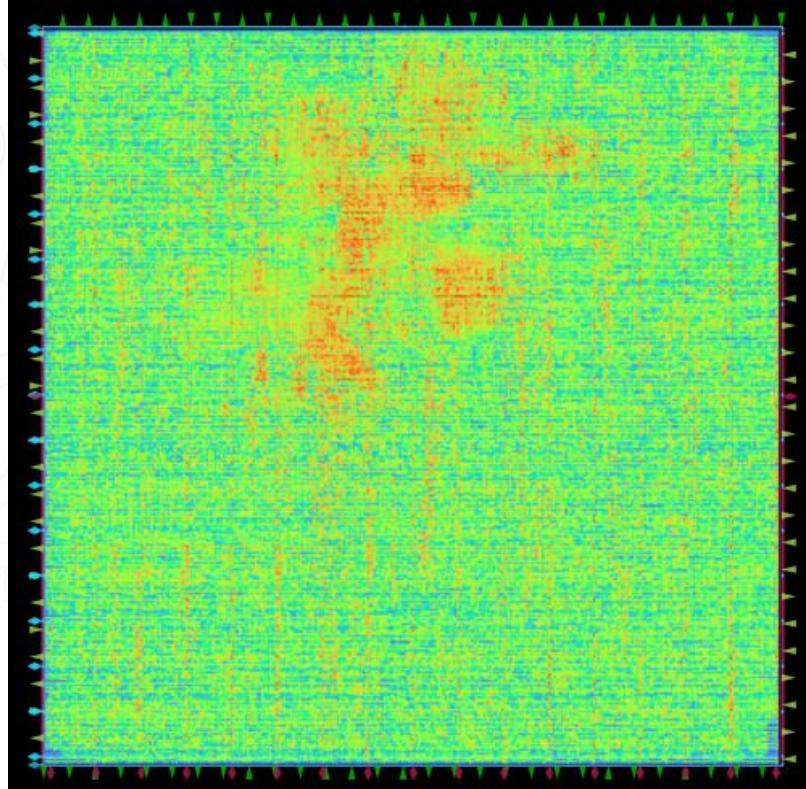
Güç Tüketimi: **124 miliWatt**

Fanout: **736**

Anten: **619**

DRC, LVS, Hold, Cap, Slew, Setup
hatasız

Kapı seviyesi testlerini geçiyor



Çip Tasarım Akışı - Openlane Config



"CLOCK_PERIOD": 25,

"PL_RESIZER_MAX_CAP_MARGIN": 85,

"GLB_RESIZER_MAX_CAP_MARGIN": 85,

"PL_RESIZER_MAX_SLEW_MARGIN": 80,

"GLB_RESIZER_MAX_SLEW_MARGIN": 80,

"PL_RESIZER_HOLD_MAX_BUFFER_PERCENT": 75,

"PL_RESIZER_SETUP_MAX_BUFFER_PERCENT": 75,

"GLB_RESIZER_HOLD_SLACK_MARGIN": 0.5,

"GLB_RESIZER_SETUP_SLACK_MARGIN": 0.5,

- Openlane Commit
dc8e6bb151df7ee2b93d7e9d6a15abf6f224db4d
(HEAD, tag: 2023.03.12)
- Skywater 130 PDK
e6f9c8876da77220403014b116761b0b2d79aab4

"PL_RESIZER_ALLOW_SETUP_VIOS": 1,

"GLB_RESIZER_ALLOW_SETUP_VIOS": 1,

"FP_SIZING": "absolute",

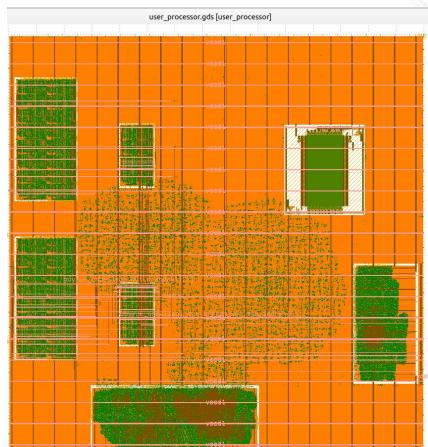
"DIE_AREA": "0 0 2500 2500",

"PL_TARGET_DENSITY": 0.34,

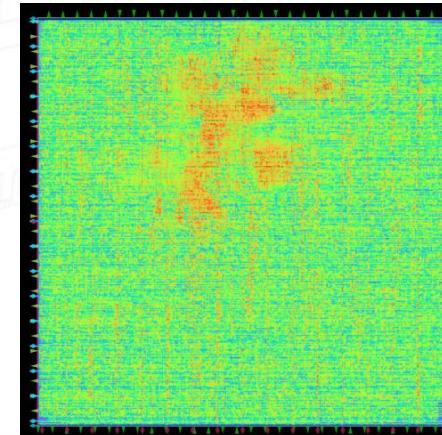
Çip Tasarım Akışı - Tasarım Karşılaştırması



- Güç Tüketimi: 5+ GigaWatt
- Frekans: 50MHz
- Anten: 1723
- Alan: 3000um x 3000um
- DRC,LVS,Hold,Cap,Setup hatasız
- Kapı seviyesi testleri yapılmadı

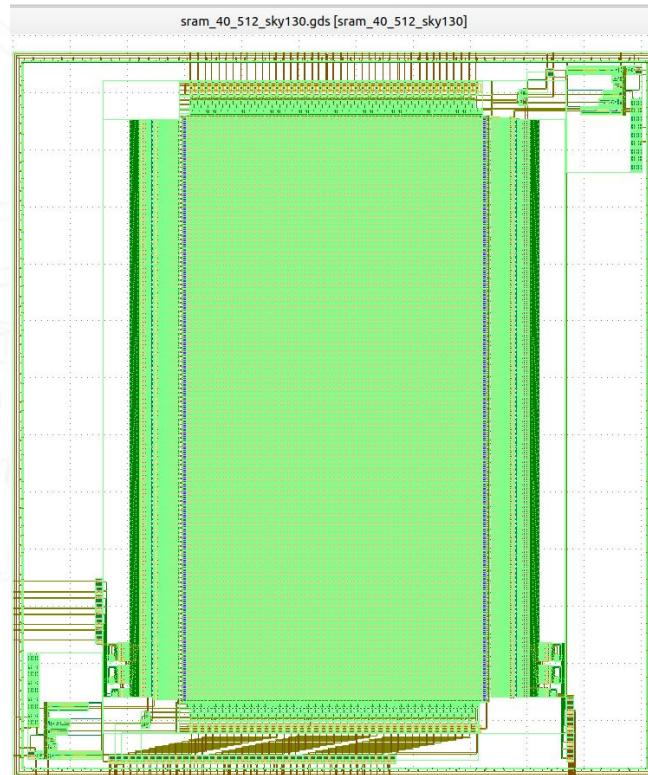


- Güç Tüketimi: 124 miliWatt
- Frekans: 40MHz
- Anten: 619
- Alan: 2500um x 2500um
- DRC,LVS,Hold,Cap,Setup hatasız
- Kapı seviyesi testlerini geçiyor
- Makrosuz (Tek top modül:
user_processor.v)



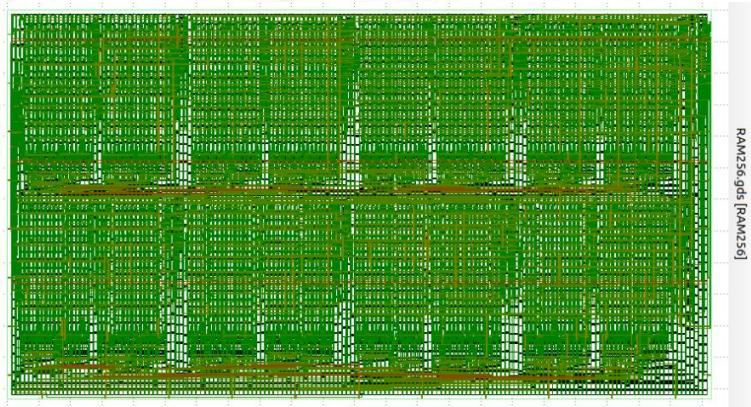
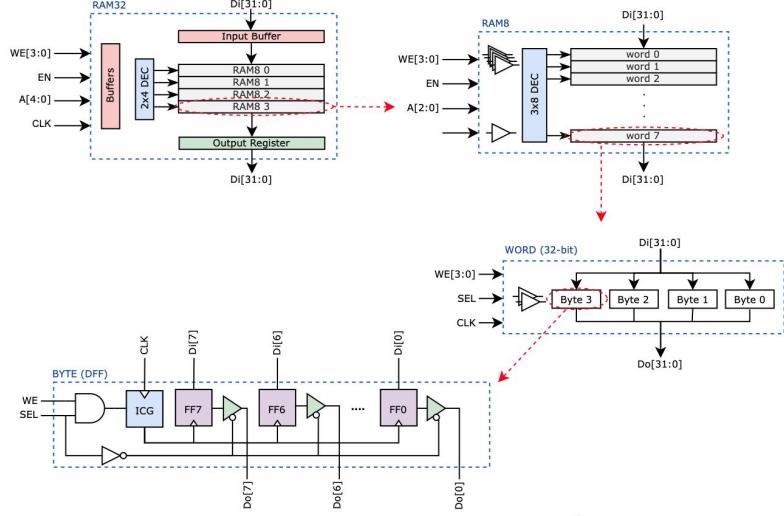
Çip Tasarım Akışı - OpenRAM

- LVS hataları.
- CAP hataları.
- Port yerlerinin uyumsuzluğu.
- Alan avantajının routing'den dolayı kullanılamaması.
- Makro olarak kullanıldığından alan avantajının, içerisindeki büyük boşluklardan dolayı daha da azalması.



Çip Tasarım Akışı - DFFRAM

- FlipFlop ve Latch tabanlı
- Senkron okuma ve yazma
- Özel cell yerleştirici (Sorunlu)



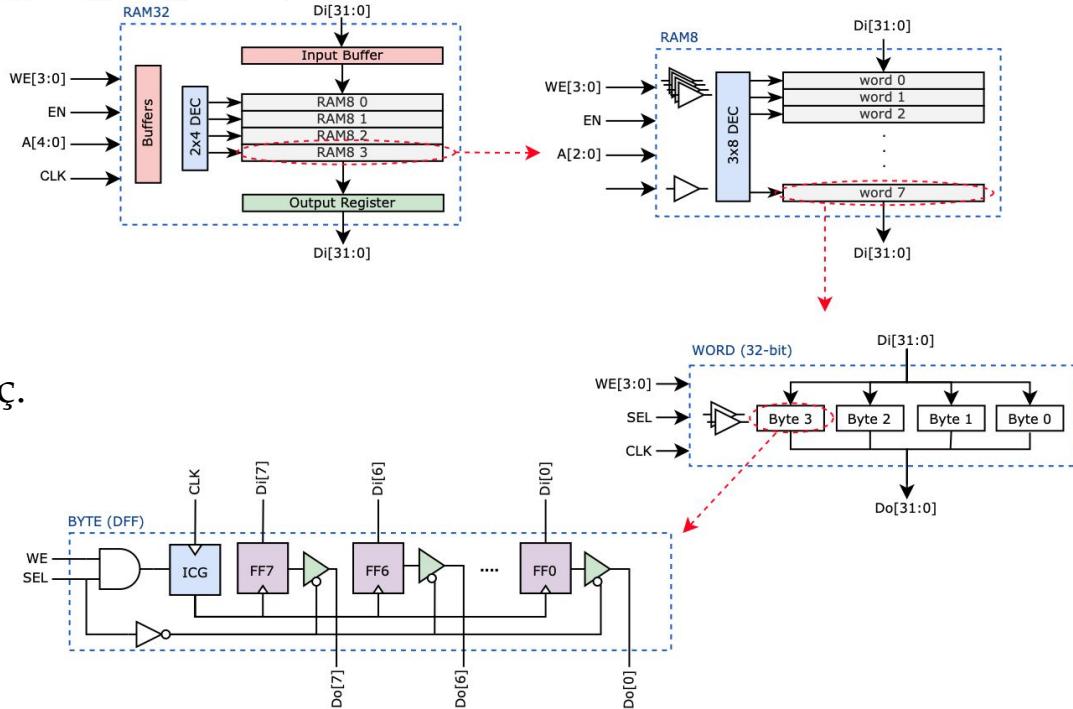
Çip Tasarım Akışı - RAM Karşılaştırma



	OpenRAM	DFFRAM	Kendi RAM'lerimiz
Alan	+++	++	+++
OpenLane	+	+++	++++
Frekans	++++	+++	+++

Çip Tasarım Akışı - Kendi RAM'lerimiz

- DFFRAM'den esinlenildi.
- Tristate yok.
- Latch yok.
- Giriş çıkışlar en basit halde.
- Timing hatası varsa buffer at geç.
- Cap hatası varsa cap at geç.
- Detaylı OpenLane kontrolü.



Çip Tasarım Akışı - Temiz Test Raporları



Magic DRC Summary:

Source: /openlane/designs/kizil_islemci/runs/
Total Magic DRC violations is 0

LVS Summary:

Source: /openlane/designs/kizil_islemci/runs/
Number of nets: 162112
Design is LVS clean.

=====
| report_power
=====

Group	Internal Power	Switching Power	Leakage Power	Total Power (Watts)
Sequential	2.00e-02	4.38e-03	4.09e-07	2.44e-02 19.6%
Combinational	4.95e-02	5.06e-02	1.32e-06	1.00e-01 80.4%
Macro	0.00e+00	0.00e+00	0.00e+00	0.00e+00 0.0%
Pad	0.00e+00	0.00e+00	0.00e+00	0.00e+00 0.0%
Total	6.95e-02	5.49e-02	1.72e-06	1.24e-01 100.0%
	55.8%	44.2%	0.0%	

=====
max slew violation count 0
max fanout violation count 736
max cap violation count 0
=====

flow_status	critical_path_ns	suggested_clock_f	suggested_clock_f	CLOCK_PERIOD	DIEAREA_mm^2	lys_total_errors
flow completed	23.36	25	40	25	6.25	0
tritonRoute_violations	Short_violations	MetSpc_violations	OffGrid_violations	MinHole_violations	Other_violations	Magic_violations
	0	0	0	0	0	0
						antenna_violations
						619

Openlane ve FPGA Optimizasyonu

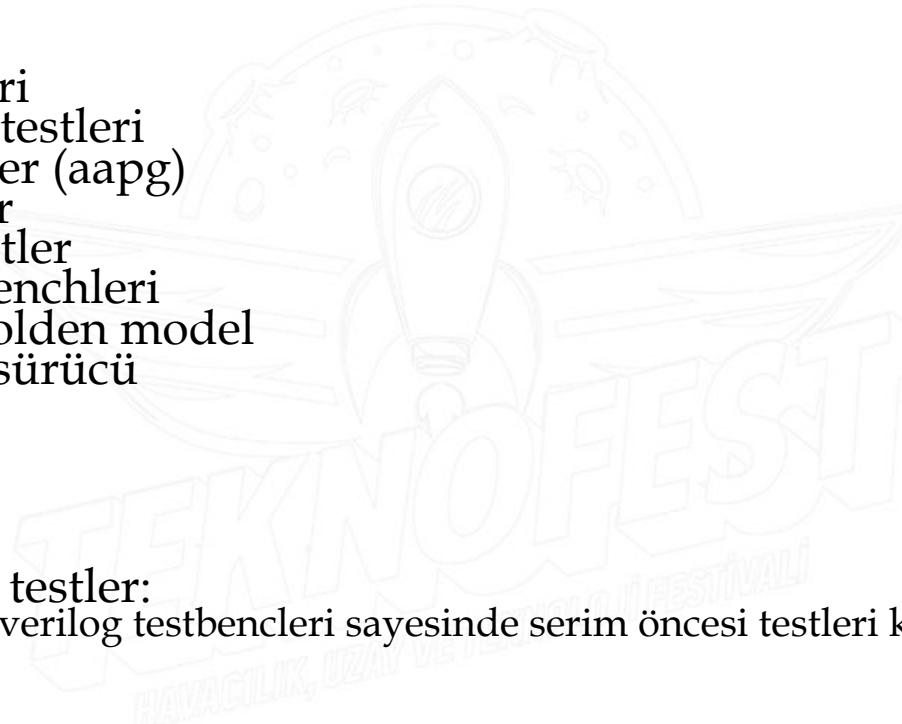


- Kapı seviyesi ve davranışsal model arası geçişler `define ile sağlanır.
- Testlerin hepsi kapı seviyesi önbelleklerle de geçiyor.

```
RAM256x8_ASYNC`GATE vffram_to_0(  
    .CLK(clk),  
    .A0(yol_A0),  
    .Di0(yol_Di0[39:32]),  
    .Do0(yolo_Do0[39:32]),  
    .WE0(yolo_EN0)  
);
```

Test

- Serim Öncesi:
 - RISC-V testleri
 - RISC-V Arch testleri
 - Random testler (aapg)
 - Benchmarklar
 - El Yapımı Testler
 - Verilog Testbenchleri
 - UART/SPI golden model
 - PWM motor sürücü
 - FPGA testleri
- Serim Sonrası:
 - Kapı seviyesi testler:
 - Cocotb ve verilog testbencleri sayesinde serim öncesi testleri kapı seviyesinde de çalıştırıldı.



Test - FPGaler



VCU108 (xcvu095-ffva2104-2-e)
125 MHz
307 CoreMark Iterations/Sec
2.5 CoreMark/MHz



Nexys A7-100T (xc7a100tcs324-1)

60 MHz

153 CoreMark Iterations/Sec
2.55 CoreMark/MHz

Vivado 2022.2



Basys3 (xc7a35tcp236-1)

60 MHz

153 CoreMark Iterations/Sec
2.55 CoreMark/MHz

Basys3 BRAM ve LUT
yetmediği için RAM
satırı 'h5000

Test - FPGALer - Utilization

FPGA Board	Frekans (MHz)	LUT	FF	BRAM	DSP	Toplam On-chip Güç (mW)
VCU108 (xcvu095-ffva2104-2-e)	125	7758	5545	128	4	1147
NEXYS A7-100T (xc7a100tcs324-1)	60	8117	5541	128	4	279
BASYS3 (xc7a35tcp236-1)	60	8194	5538	*32	4	240

Test - FPGALer - Utilization - Modüller

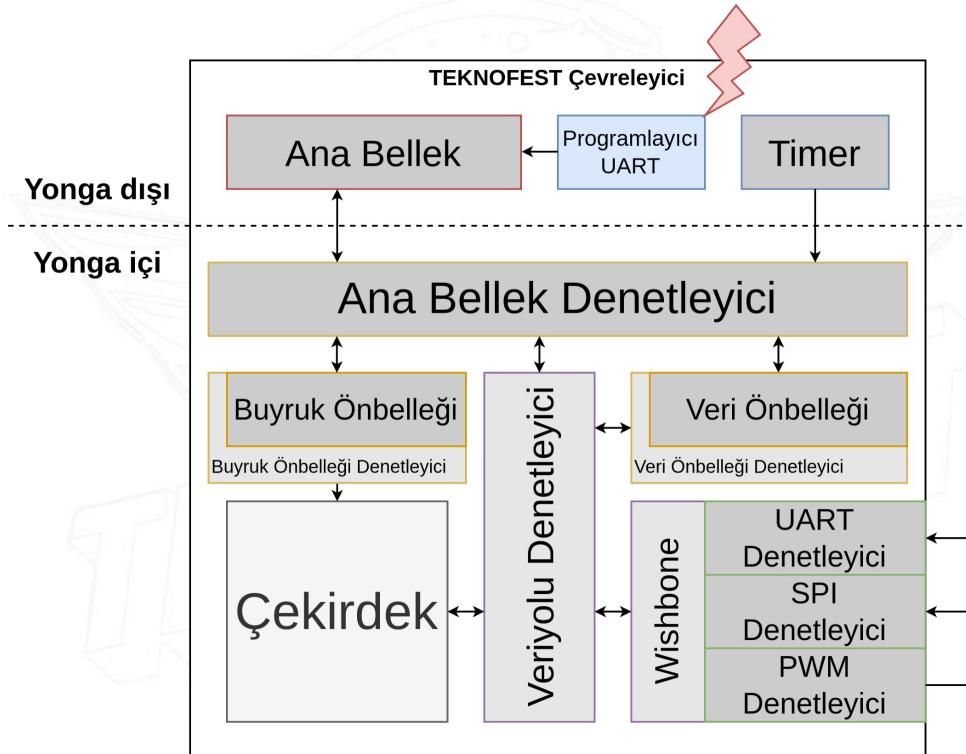


Modüller	LUT	FF	DSP
Belleksiz İşlemci	6969	5172	4
Çekirdek	3505	2826	4
UART	108	112	0
SPI	913	944	0
PWM	1092	647	0

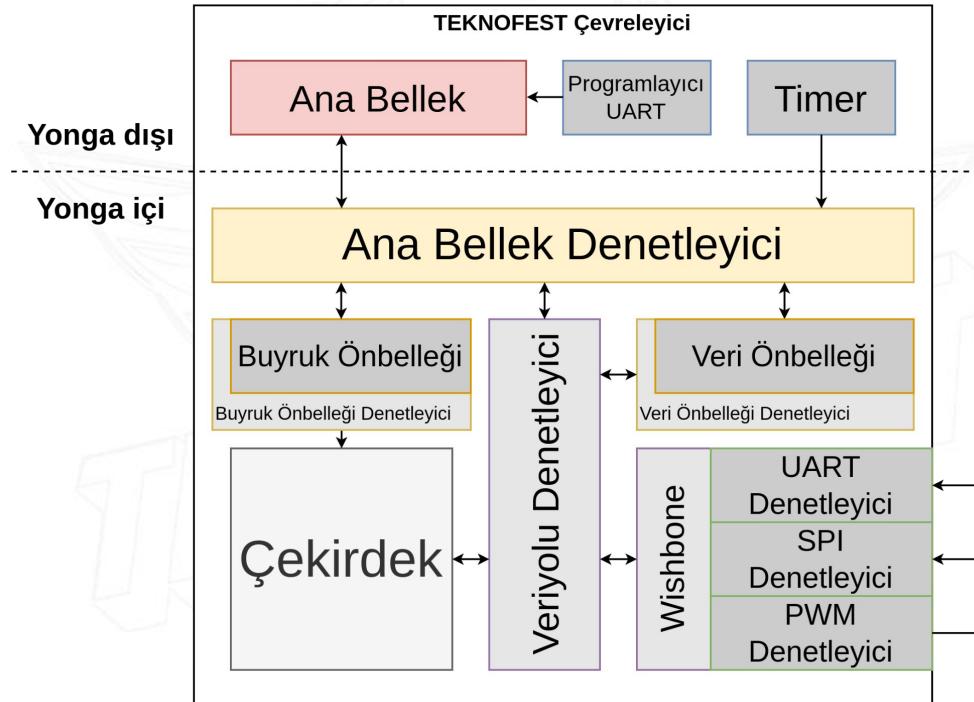
Test - Programlama ve Test Ortamı



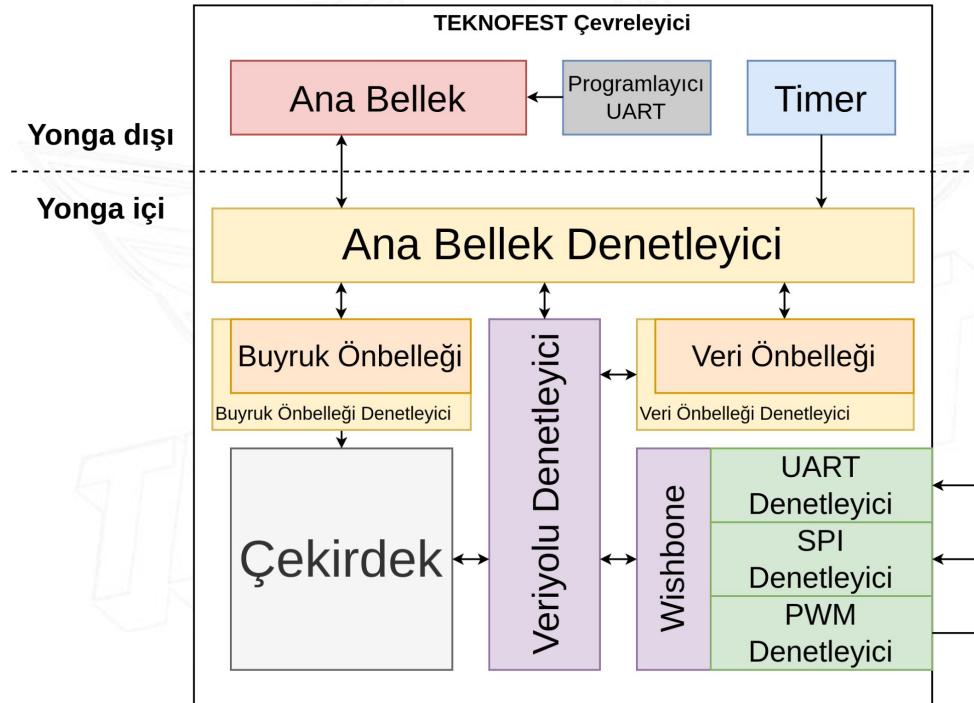
Test - Adım Adım Programlama



Test - Adım Adım Programlama



Test - Adım Adım Programlama



Test - RISCV ve RISCV Arch Testleri

- Python + Cocotb

[TEST] auipc passed	[TEST] srl passed	[TEST] mulhu passed	[TEST] pkg passed
[TEST] jal passed	[TEST] sra passed	[TEST] mulhsu passed	[TEST] sladd passed
[TEST] jalr passed	[TEST] slt passed	[TEST] div passed	[TEST] cntz passed
[TEST] lui passed	[TEST] sltu passed	[TEST] divu passed	[TEST] cntp passed
[TEST] andi passed	[TEST] srli passed	[TEST] rem passed	[TEST] conv passed
[TEST] ori passed	[TEST] srai passed	[TEST] remu passed	
[TEST] xori passed	[TEST] sub passed	[TEST] lw passed	
[TEST] addi passed	[TEST] bgeu passed	[TEST] lh passed	
[TEST] slli passed	[TEST] bltu passed	[TEST] lb passed	
[TEST] slti passed	[TEST] blt passed	[TEST] lbu passed	
[TEST] sltiu passed	[TEST] bne passed	[TEST] lhu passed	
[TEST] and passed	[TEST] beq passed	[TEST] sw passed	
[TEST] sll passed	[TEST] bge passed	[TEST] sb passed	
[TEST] xor passed	[TEST] add passed	[TEST] sh passed	
[TEST] or passed	[TEST] mul passed	[TEST] hmdst passed	
	[TEST] mult passed	[TEST] rvrs passed	

Test - Rastgele Testler



- COCOTB + AAPG + SPIKE
 - Trace ve signature karşılaştırması
 - Random Coverage Testleri
 - IMC eklentili binlerce rastgele buyruk
 - Bug çözümü

Test - UART

- UART RX Testleri
 - UART TX Testleri
 - UART Buffer Stres Testleri
 - Golden Model:
https://github.com/alexfo_ncich/cocotbext-uart

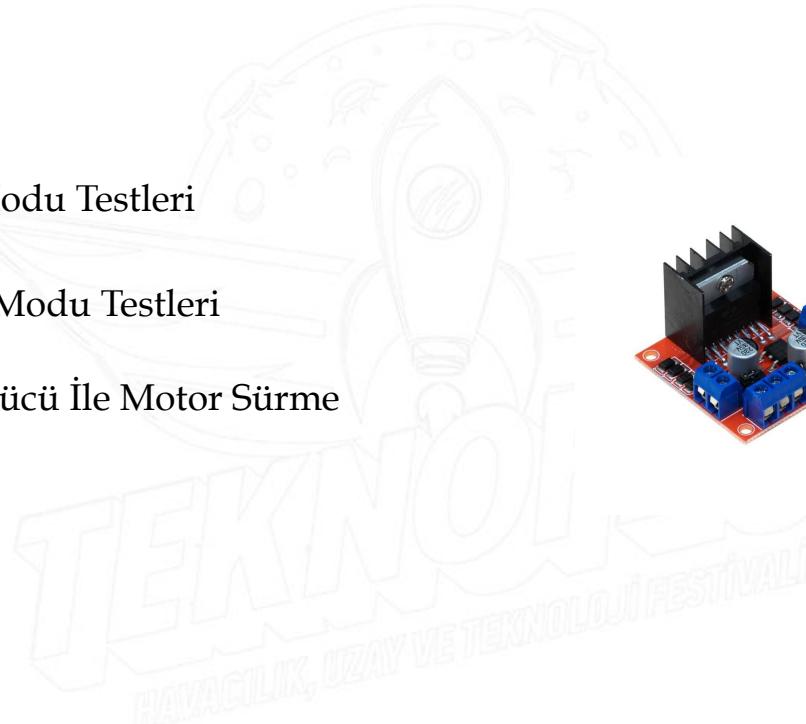
Test - SPI

- SPI Alıcı Testleri
- SPI Verici Testleri
- SPI Arduino Slave Testleri
- Golden Model: <https://github.com/schang412/cocotbext-spi>



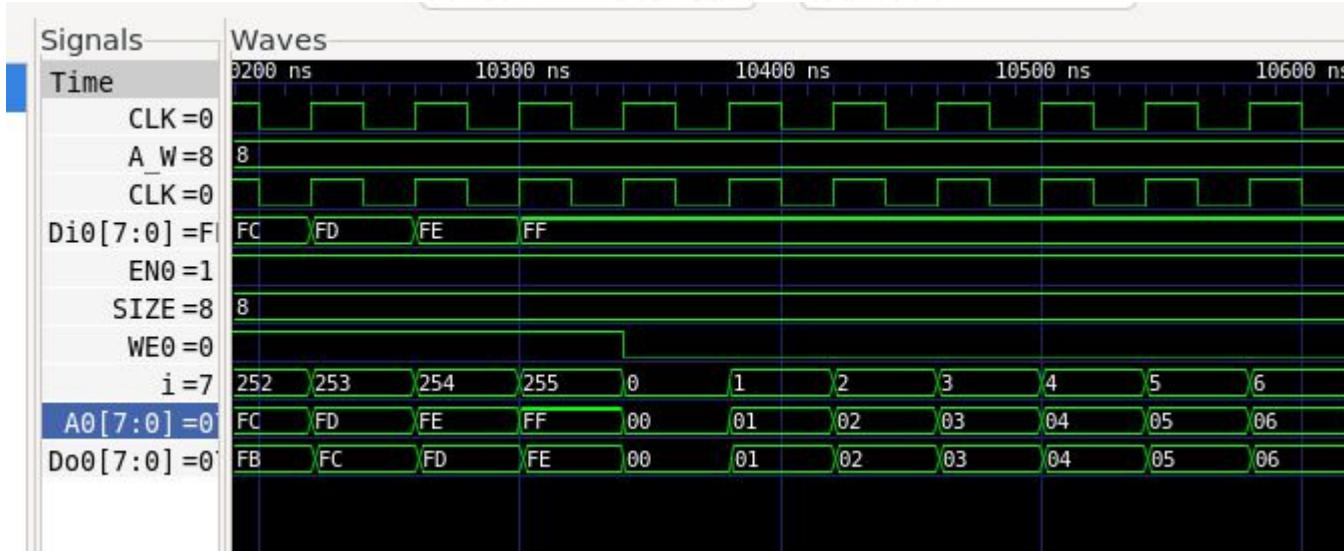
Test - PWM

- PWM Standart Modu Testleri
- PWM Kalp Atışı Modu Testleri
- L298N Motor Sürücü İle Motor Sürme



Test - El Yapımı RAM'ler

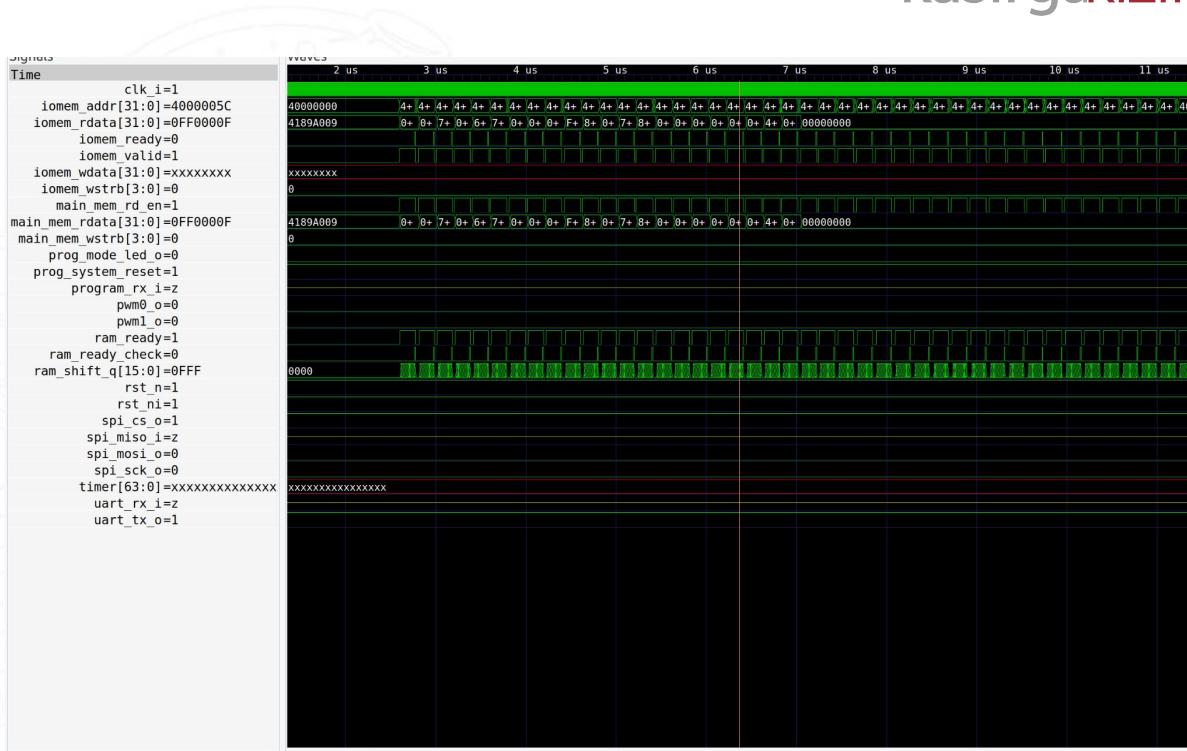
- El yapımı RAM'ler bütün testleri (kapı seviyesi dahil) geçiyor.
- DFFRAM'in testlerini bile geçiyor.



Test - Kapı Seviyesi (Serim Sonrası Test)



- Teknofest wrapper verilog kodu, saat ve reset sinyalinin içerisinde üretilmesi ve belleğe initial dosyası verilmesi ile gate-level testbench'e çevrilmiştir.
- Riscv testleri ile icarus üzerinden denenmiş ve çalıştığı görülmüştür.



Test - Coremark - Nexys A7 (60 MHz)



2000 iterasyon

153 CoreMark Iterations/Sec
2.55 CoreMark/MHz

```
parameters
 2K performance run parameters for coremark.
CoreMark Size      : 666
Total ticks       : 825907498
Total time (secs) : 13
Iterations/Sec    : 153
Iterations        : 2000
Compiler version  : GCC12.2.0
Compiler flags     : -O2 -mcmode=medany -static -std=gnu99 -fno-common -nostdlib
                     -nostartfiles -fno-builtins -ffunction-sections -lm -lgcc -T riscv32-baremetal/li
                     nkdefn.lds -DPERFORMANCE_RUN=1
Memory location   : STACK
seedcrc           : 0xe9f5
[0]crclist        : 0xe714
[0]crcmatrix      : 0x1fd7
[0]crcstate        : 0x8e3a
[0]crcfinal        : 0x4983
Correct operation validated. See README.md for run and reporting rules.
```

Test - Coremark - VCU108 (125 MHz)



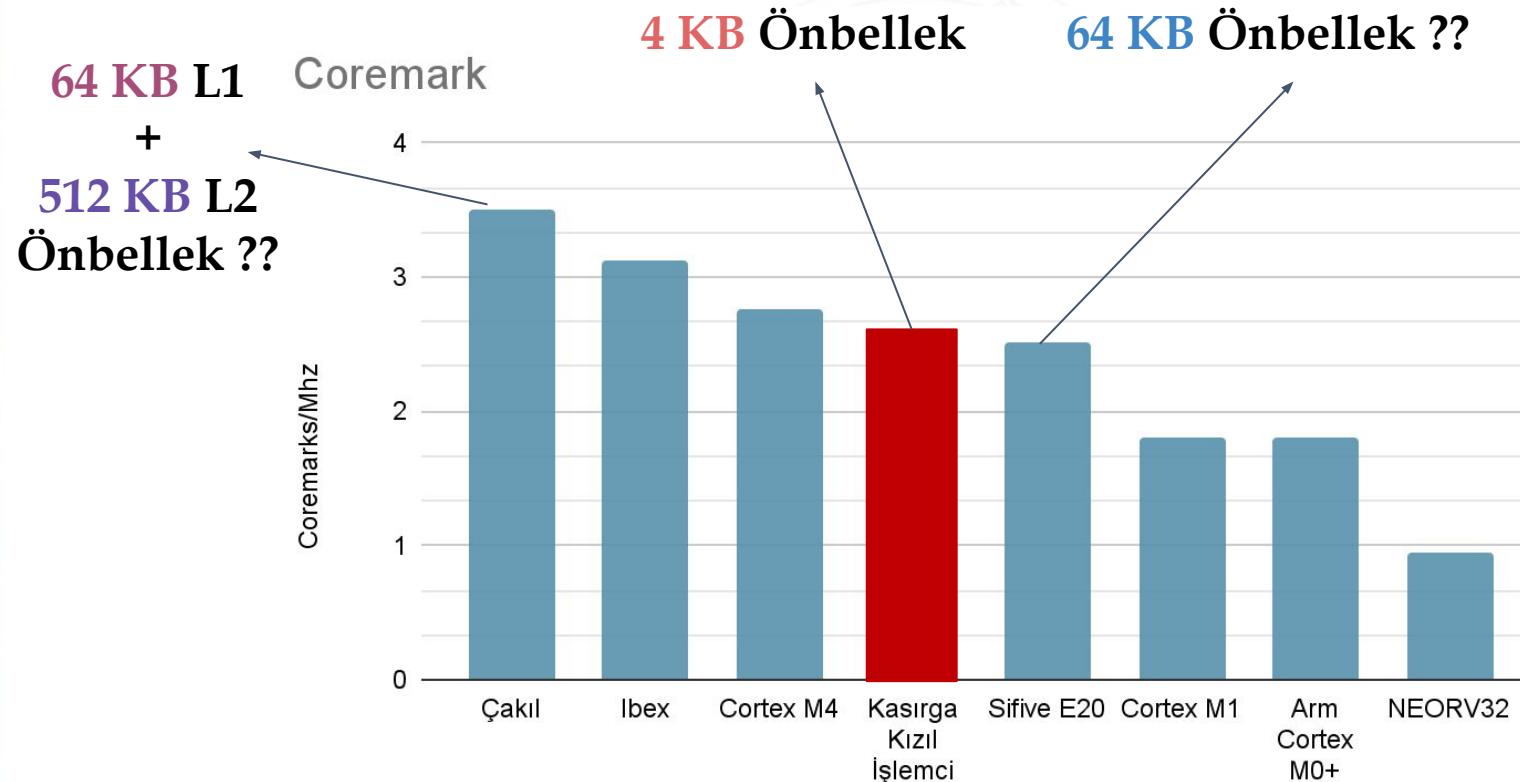
4000 iterasyon

307 CoreMark Iterations/Sec

2.5 CoreMark/MHz

```
2K performance run parameters for coremark.
CoreMark Size      : 666
Total ticks       : 1651804755
Total time (secs): 13
Iterations/Sec    : 307
Iterations        : 4000
Compiler version  : GCC12.2.0
Compiler flags     : -O2 -mcmodel=medany -static -std=gnu99 -fno-common -nostdlib
                     -nostartfiles -fno-builtin -ffunction-sections -lm -lgcc -T riscv32-baremetal/li
                     nkdenelld -DPERFORMANCE_RUN=1
Memory location   : STACK
seedcrc           : 0xe9f5
[0]crcclist       : 0xe714
[0]crcmatrix      : 0x1fd7
[0]crcstate        : 0x8e3a
[0]crcfinal        : 0x65c5
Correct operation validated. See README.md for run and reporting rules.
```

Test - CoreMark Karşılaştırma



Test - CoreMark - Veri Ön belleği Hit Rate



50 ms boyunca
CoreMark çalıştırılarak
Veri Ön belleğinin hit
rate'si yanda görüldüğü
gibi ölçülmüştür.

CoreMark'ın zamanda
yerellik özelliğinden
çokça faydalananması
nedeni ile hit rate %97.8
olarak ölçülmüştür.

```
# [TEST] coremark FAILED TIMEOUT
# current PC: 1073748860
# Load Hit sayac: 13915
# Bellek Load sayac: 338
# Store Hit sayac: 4570
# Bellek Store sayac: 71
# 50000055.00ns INFO cocotb.regression
```

Test - CoreMark - Değişikliklerin Etkisi



VCU108 → 100 MHz

Yapılan Tasarım Değişikliği	Eski CoreMark Skoru (Coremark/Sec)	Yeni Coremark Skoru (Coremark/Sec)	Performans Artışı
Çarpma İşlemine Yönlendirme Eklenmesi	166	181	%9.0
Veri Önbelleğinin Tek Yolu OpenRam'den İki Yolu Kombinasyonel Flip-Flop Devresine Çevirilmesi	181	237	%30.9
BTB yerine Toplayıcı Tabanlı Atlama Adresi Hesaplayıcı Kullanılması	237	255	%7.5
Buyruk Önbelleğinin Tek Yoldan İki Yolluya Çevrilmesi	255	255	%0.8

Test - Dhryystone - Nexys A7 (60 MHz)



2 milyon iterasyon

123227 Dhryystone per second
8 Microseconds for one run
through Dhryystone

```
Dhrystone Benchmark, Version 2.1 (Language: C)
Program compiled without 'register' attribute
Execution starts, 2000000 runs through Dhrystone
Execution ends

Final values of the variables used in the benchmark:

Int_Glob:      5
    should be: 5
Bool_Glob:     1
    should be: 1
Ch_1_Glob:     A
    should be: A
Ch_2_Glob:     B
    should be: B
Arr_1_Glob[8]: 7
    should be: 7
Arr_2_Glob[8][7]: 2000010
    should be: Number_Of_Runs + 10
Ptr_Glob→
    Ptr_Comp: 1073758944
        should be: (implementation-dependent)
    Discr:   0
    should be: 0
    Enum_Comp: 2
        should be: 2
    Int_Comp: 17
        should be: 17
    Str_Comp: DRYSTONE PROGRAM, SOME STRING
        should be: DRYSTONE PROGRAM, SOME STRING
Next_Ptr_Glob→
    Ptr_Comp: 1073758944
        should be: (implementation-dependent), same as above
    Discr:   0
    should be: 0
    Enum_Comp: 1
        should be: 1
    Int_Comp: 18
        should be: 18
    Str_Comp: DRYSTONE PROGRAM, SOME STRING
        should be: DRYSTONE PROGRAM, SOME STRING
Int_1_Loc:      5
    should be: 5
Int_2_Loc:      13
    should be: 13
Int_3_Loc:      7
    should be: 7
Enum_Loc:       1
    should be: 1
Str_1_Loc:      DRYSTONE PROGRAM, 1'ST STRING
    should be: DRYSTONE PROGRAM, 1'ST STRING
Str_2_Loc:      DRYSTONE PROGRAM, 2'ND STRING
    should be: DRYSTONE PROGRAM, 2'ND STRING

Microseconds for one run through Dhrystone: 8
Dhrystones per Second: 123227
```

Test - Dhryystone - VCU108 (125 MHz)



3 milyon iterasyon

349232 Dhryystone per second
2 Microseconds for one run
through Dhryystone

```
Dhrystone Benchmark, Version 2.1 (Language: C)
Program compiled with 'register' attribute
Execution starts, 3000000 runs through Dhrystone
Execution ends
Final values of the variables used in the benchmark:
Int_Glob:      5
              should be: 5
Bool_Glob:     1
              should be: 1
Ch_1_Glob:    A
              should be: A
Ch_2_Glob:    B
              should be: B
Arr_1_Glob[8]: 7
              should be: 7
Arr_2_Glob[8][7]: 3000010
              should be: Number_Of_Runs + 10
Ptr_Glob->
  Ptr_Comp:    1141915323
              should be: (implementation-dependent)
Discr:        0
              should be: 0
Enum_Comp:   2
              should be: 2
  Int_Comp:   17
              should be: 17
  Str_Comp:  DHRYSTONE PROGRAM, SOME STRING
              should be: DHRYSTONE PROGRAM, SOME STRING
Next_Ptr_Glob->
  Ptr_Comp:    1141915323
              should be: (implementation-dependent), same as above
  Discr:        0
              should be: 0
  Enum_Comp:   1
              should be: 1
  Int_Comp:   18
              should be: 18
  Str_Comp:  DHRYSTONE PROGRAM, SOME STRING>D
              should be: DHRYSTONE PROGRAM, SOME STRING
  Int_1_Loc:   5
              should be: 5
  Int_2_Loc:   13
              should be: 13
  Int_3_Loc:   7
              should be: 7
  Enum_Loc:   1
              should be: 1
  Str_1_Loc:  DHRYSTONE PROGRAM, 1'ST STRING
              should be: DHRYSTONE PROGRAM, 1'ST STRING
  Str_2_Loc:  DHRYSTONE PROGRAM, 2'ND STRING
              should be: DHRYSTONE PROGRAM, 2'ND STRING

Microseconds for one run through Dhrystone: 2
Dhrystones per Second: 349232
```

Test - Özel Buyruklar



- RISC-V test makroları ile özel buyrukları destekleyen testler yazıldı.
- Assembly inline Tütel testleri de denendi.

TEST_RR_DEST_BYPASS(10, 1, hmdst, 0xb, 0x95ef299f, 0x1065418b);

TEST_RR_SRC12_BYPASS(16, 1, 1, hmdst, 0xb, 0x95ef299f, 0x1065418b);

TEST_RR_SRC21_BYPASS(22, 1, 1, hmdst, 0xb, 0x95ef299f, 0x1065418b);

TEST_RR_ZEROSRC1(24, hmdst, 0x14, 0x95ef299f);

TEST_RR_ZEROSRC2(25, hmdst, 0x14, 0x95ef299f);

TEST_RR_ZEROSRC12(26, hmdst, 0x0);

TEST_RR_ZERODEST(27, hmdst, 0x11111111, 0x22222222);

TEST_RR_OP(28, hmdst, 0xe , 0x12345678, 0x87654321);

Test - LLVM - Özel Buyruklar



kasırgakızıl

```
unsigned int rvrs(unsigned int reg1){  
    return ((0xff & reg1)<<24)+  
           ((0xff & (reg1>>8))<<16)+  
           ((0xff & (reg1>>16))<<8)+  
           ((0xff & (reg1>>24)));  
}
```

```
00010210 <rvrs>:  
 10210: 13 01 01 ff  addi   sp, sp, -16  
 10214: 23 26 11 00  sw     ra, 12(sp)  
 10218: 23 24 81 00  sw     s0, 8(sp)  
 1021c: 13 04 01 01  addi   s0, sp, 16  
 10220: 23 2a a4 fe  sw     a0, -12(s0)  
 10224: 03 25 44 ff  lw     a0, -12(s0)  
 10228: 13 55 85 6b  <unknown>  
 1022c: 03 24 81 00  lw     s0, 8(sp)  
 10230: 83 20 c1 00  lw     ra, 12(sp)  
 10234: 13 01 01 01  addi   sp, sp, 16  
 10238: 67 80 00 00  ret
```

```
0001023c <rvrs>:  
 1023c: 13 01 01 ff  addi   sp, sp, -16  
 10240: 23 26 11 00  sw     ra, 12(sp)  
 10244: 23 24 81 00  sw     s0, 8(sp)  
 10248: 13 04 01 01  addi   s0, sp, 16  
 1024c: 23 2a a4 fe  sw     a0, -12(s0)  
 10250: 03 25 44 ff  lw     a0, -12(s0)  
 10254: 93 55 85 01  srli   a1, a0, 24  
 10258: 13 56 85 00  srli   a2, a0, 8  
 1025c: b7 06 01 00  lui    a3, 16  
 10260: 93 86 06 f0  addi   a3, a3, -256  
 10264: 33 76 d6 00  and   a2, a2, a3  
 10268: b3 e5 c5 00  or    a1, a1, a2  
 1026c: 13 16 85 00  slli   a2, a0, 8  
 10270: b7 06 ff 00  lui    a3, 4080  
 10274: 33 76 d6 00  and   a2, a2, a3  
 10278: 13 15 85 01  slli   a0, a0, 24  
 1027c: 33 65 a6 00  or    a0, a2, a0  
 10280: 33 e5 a5 00  or    a0, a1, a0  
 10284: 03 24 81 00  lw     s0, 8(sp)  
 10288: 83 20 c1 00  lw     ra, 12(sp)  
 1028c: 13 01 01 01  addi   sp, sp, 16  
 10290: 67 80 00 00  ret
```

12 buyruktan tek buyruğa

İş Planı

- İş paketleri, iş planında öngörülen sürelerde tamamlanmıştır.

GÖREV TANIMI	ÖTR Aşaması		DTR Aşaması		Final Aşaması	
	14.10.22 - 05.11.22	6.11.22 - 11.12.22	12.12.22 - 22.01.23	23.01.23 - 15.03.23	16.03.23 - 02.04.23	03.04.23 - 17.04.23
Sistemin Teorik Tasarımı Ön Tasarım Raporunun Hazırlanması ve Teslimi Verilog Tasarımının Gerçekleştirilmesi Verilog Modüllerinin Testi ve Doğrulanması İşlemeçinin FPGA Üzerinde Demo Edilmesi OpenLane ve OpenRAM Çalışmaları Detay Tasarım Raporunun Hazırlanması ve Teslimi Tasarımın Nihai Hale Getirilmesi Final Sunumunun Hazırlanması						

Kaynakça

- [1] Kanamori, Takuto, Hiromu Miyazaki, and Kenji Kise. "RVCoreP-32IC: A high-performance RISC-V soft processor with an efficient fetch unit supporting the compressed instructions." *arXiv preprint arXiv:2011.11246* (2020).
- [2] Temel, Mertcan, Anna Slobodova, and Warren A. Hunt Jr. "Automated and scalable verification of integer multipliers." *Computer Aided Verification: 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21–24, 2020, Proceedings, Part I*. Cham: Springer International Publishing, 2020.
- [3] https://github.com/SerdarUnal132/pwm_openmpw
- [4] https://github.com/tdene/synth_optadders
- [5] [https://wwwcdn.aselsan.com/api/file/CAKIL_TR-\(1\)-\(1\).pdf](https://wwwcdn.aselsan.com/api/file/CAKIL_TR-(1)-(1).pdf)
- [6] <https://www.sifive.com/cores/e20>



TEŞEKKÜRLER