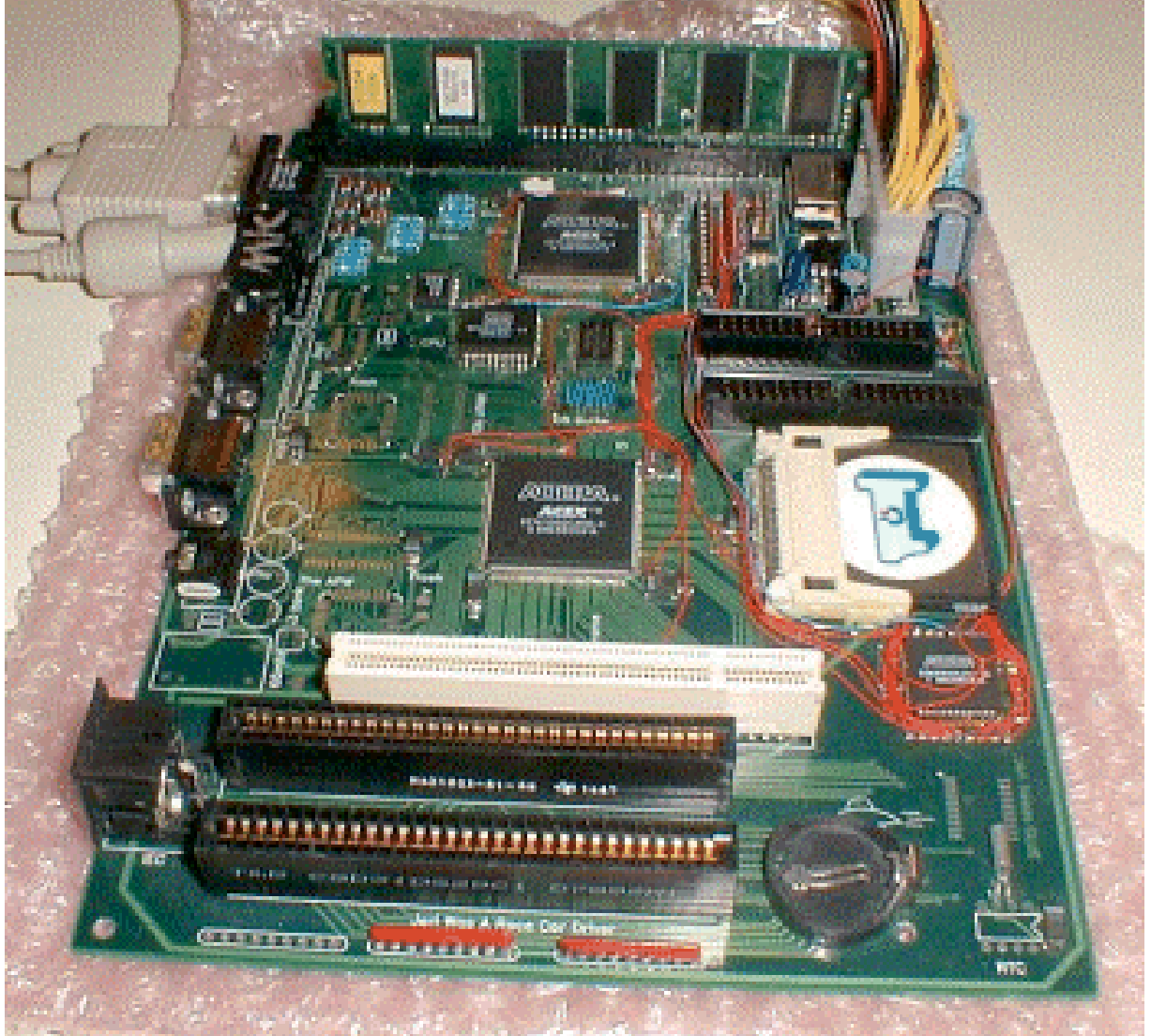


C64 TÜRKİYE

SAYI : #S02

MART 2003

BİLGİ PAYLAŞTIKÇA ARTAR



**COMMODORE ONE
ASSEMBLER KURSU 2
PROGRAM KÖŞESİ
BELLEK HARİTASI 2**

Hazırlayan ve sahibi : İSMAİL "HADES" ŞAHİN
Katkıda bulunanlar : DENİZ CAN ÇELİK
Kapak Resmi : CommodoreOne anakartı

İÇİNDEKİLER

BU SAYFA	2
HABERLER	3
COMMODORE ONE	6
ASSEMBLER KURSU – 2	9
PROGRAM KÖŞESİ	13
BELLEK HARİTASI – 2	16

- 1- *Dergi için COMMODORE 64 ile ilgili her türlü yazı, resim, program, oyun tanıtımı, ipuçları, hile ve ufak-tefek rutinler gibi malzemeler kabul edilir.*
- 2- *Yukarıdaki malzemelerin çalışır durumda olduğundan emin olunmalıdır.*
- 3- *Gönderilen malzemeler dergide mutlaka yayınlanacak diye herhangi bir şart yoktur.*
- 4- *Malzemeyi gönderenin adı-soyadı-e-mail adresi-oturduğu şehir ve kendisini tanıtan kısa bir not yazması tercih edilir.*
- 5- *Derginin editörü gönderilen malzemelerde değişiklik, düzenleme vs yapabilir.*
- 6- *Gönderilen malzemelerin sorumluluğu gönderene aittir.*
- 7- *Hiç bir şekilde siyasi, dini yazılara yer verilmeyecektir. Ayrıca kişilere ve şirketlere yönelik hakaret, aşağılama, küfür gibi yazılar yayınlanmayacaktır.*
- 8- *Emeğe saygı gereği kaynak göstermek şartıyla herkes dergideki yazılardan alıntı yapabilir.*
- 9- **BU DERGİ KESİNLİKLE TİCARİ AMAÇLI DEĞİLDİR VE PARA İLE SATILAMAZ.**
- 10- *Dergiyi okuyan herkes bütün bu maddeleri kabul etmiş sayılır. (Nasıl olacaksa?)*

EDİTÖRÜN KLAVYESİ

Uzun bir aradan sonra herkese merhaba.....
Dergimiz C64 TÜRKİYE her ne kadar aylık olarak gözüke de tamamen kişisel çabalarla hazırlandığı için elimizde olmayan teknik nedenlerden dolayı gecikmeler yaşanmaktadır. ☺☺☺

Gecikmenin sebeplerinin başında bilgisayar ile uğraşacak zaman bulamayışım gelmektedir. Daha da kötüsü modemin bozulması ve lanet olası bir virüsün bazı dosyalara bulaşması sonucu bana kafayı yedirtmesidir. Allah'tan işyerindeki PC'de dergi çalışmalarının yedekleri vardı da 2. sayıyı sil baştan yapmaktan kurtuldum.

Bu kadar bahaneden sonra bu sayıya bir göz atalım. Her sayıyı bir öncekinden daha iyi yapmak için biraz zamana ve biraz da katkıya gerek var.

Fark ettiğiniz gibi kapak yazıları renkli oldu ve heyecanla ve sabırla beklenen COMMODORE ONE 'dan bir resim koyduk. Bu resim aynı zamanda derginin kapak konusu oldu.

Ayrıca HABERLER sayfaları ile C64 dünyasında olup bitenleri sizlere iletmeye çalıştık. Haberlerde göreceğiniz gibi C64 ile iki dergi daha çıkmış durumda. Almanca olan LOTEK64 ve İngilizce COMMODORE CURRENTS dergileri. Ne dersiniz, acaba dev uyanıyor mu ?

Buradan COMMODORE ONE yazısını ve HABERLER'in %99.9'unu hazırlayan DENİZ CAN ÇELİK'e teşekkür ederim.

ASSEMBLER KURSU 2. bölümü ile devam ediyor. Assembler kursu 4-5 sayı daha devam edecek gözüküyor.

PROGRAM KÖŞESİ'ndeki programın hikayesi ise şöyle... Evin salonunda can sıkıntısından volta atarken birden aklıma geldi ve 2 saat içinde gördüğünüz program hazır hale geldi. Her neyse devam edelim.

BELLEK HARİTASI nihayet bitti ama bende bittim. Bir sonraki sayıda bellek haritasından boşalan sayfalar için çok çalışmam lazım. Oyun tanıtımı veya daha fazla program köşesi olabilir.

Yeni köşemiz olan GEYİK KÖŞESİ ise tamamen sizlerden gelecek olan mavra olay, karikatür, yazı vs ile hazırlanacaktır (Tabii ki C64 ile ilgili (Bu arada editörü editlediğimin farkında mısınız!! -Deniz)).

İlgilenenlere duyurulur.

CommodoreOne çevirisinde hatalar bulunabilir (Eğer farkederseniz bildirin). Eğer bir aksilik çıkmazsa COMMODORE ONE ile ilgili daha ayrıntılı bir yazıyı 3. sayıda vermeyi planlıyoruz.

Bana ulaşmak için

1 – HADES6510@HOTMAIL.COM

2 – HADES6510@YAHOO.COM e-maillerini ve

3 – <http://hades6510.sitemynet.com> 'daki ziyaretçi defterini kullanabilirsiniz.

Bir sonraki sayıda buluşabilmek dileğiyle.....
Herkese bol klavyeli günler....

HABERLER

Maarssen Commodore Show

21 Aralık'ta Hollanda'nın Maarssen kentinde bir Commodore Show düzenlendi. Hollanda Commodore kullanıcıları grubunun düzenlediği show'a, C=1'in yaratıcısı Jeri Ellsworth de katıldı. Hatta show sırasında bir Hollanda radyosu CommodoreOne ile ilgili bir röportaj da yaptı. Röportajın çoğu Flemenkçe olmasına rağmen, Jeri Ellsworth ile konuşulan bölüm İngilizce (mp3, 5.68 MB, 8 dk 16 sn.). Röportajı <http://nostalgia.procaz.net/> adresinden indirip dinleyebilirsiniz. Hollanda Commodore kullanıcıları grubunun düzenlediği son show 15 Şubat 2003 Cumartesi günü yapıldı. Grubun internet sitesi <http://commodore-gg.hobby.nl/> adresinde. Bu arada röportajın İngilizce kısmının çevirisi de aşağıda:

- Jeri Ellsworth, burada çok popülersiniz.
- Ben... bilmiyorum, sanırım. :)
- Bunu (CommodoreOne'i) niçin yaptınız?
- Ben her zaman bir Commodore 64 hayranıydım. Babamın bana aldığı ilk bilgisayarımı. 1983 gibiydi. Bana bunu aldı ve kendimi programlamaya ve küçük video oyunlarını oynamaya adanmış.
- Bunu kendiniz mi yaptınız?
- Evet. Hemen hemen. Diğer insanlardan bilgi aldım ve ilk Commodore bilgisayarlarının tasarımıyla uğraşmış bazı Commodore mühendisleriyle konuştum ki, bana ne yapmam gerektiği hakkında ve bu çiplerin nasıl çalıştığı hakkında yardım ettiler.
- ... ve şimdi modern bir anakart haline gelmiş...
- C=1'i elimden geldiği kadar modern anakartlara benzetmeye çalıştım. Onu gerçek bir PC kasasının içine koyabilir, PC klavyesi ve PC disket sürücülerıyla kullanabilirsiniz. Hepsi bir bilgisayarcıdan kolayca bulunabilecek şeyler.
- Peki bu anakartla planlarınız nedir?
- Bu konuda gerçekten C64 kullanıcılarına bağlıyım. Diğer kişiler bunun için konfigürasyonlar yazabilir. Yani umuyorum ki insanlar bununla çalışacak şeyler yapacaklardır.
- Bunu satmaya çalışacak mısınız?
- Evet, ama önce ilk kartların üretimini tamamlamamız ve göndermeye hazır olmamız lazım.
- Bu 2003'ün yeni Commodore'u mu?
- Evet... 20 yıl sonra!

CommodoreOne Televizyonda!

16 Aralık 2002'de Alman 3Sat kanalı C=1 ile ilgili kısa bir haber yayınladı. C=1'in yakın plan çekimlerinin yer aldığı filmde Jeri Ellsworth, C=1'e bir CompactFlash kartı yerleştiriyor (Almanca, Mpeg / Real Media, 29 sn.). İşte spikerin söyledikleri:

"Eski bilgisayar meraklılarının heyecanla beklediği yeni bilgisayar, kameranın karşısında: C=1! C64 gibi eski bilgisayarları emule edebilen bir ATX anakart... Bir CompactFlash kartı, işletim sistemini içermesinin yanında donanımı da kontrol ediyor. Burada gösterilen prototip 249 Euro civarında ve şubat 2003 civarında piyasaya çıkacak."

<http://vicekb.de.vu/c1>

LUnix

64'ünde Unix kullanmak isteyen var mı? Bu fikir aslında yeni değil ama benim yeni haberim oldu. LUnix, C64'ler (ve tabii ki C64 modunda çalışan C128'ler) için bir işletim sistemi. Hayır hayır, Linux'la alakası yok. Açılımı "Little Unix", yani küçük Unix. LUnix, ayrıca bir donanım gerektirmiyor. 32 işe kadar multitasking yapabilme yeteneği var. Tabii, multitasking yaptığı programların 64 KB'lık hafızaya sığması lazım, RAM kartuşu (en azından şimdilik) desteklenmiyor. Program "küçük" oluşundan dolayı Unix'in tüm özelliklerine sahip değil. Ama cd, rm, ls gibi temel Unix komutlarını kullanma imkanı var.

<http://hld.c64.org/poldi/lunix/lunix.html>

```
e> mem
Num/processes: 4 stacks: 4
Num/pipes : 6 code: 20
Num/driver : 1 descr.: 1
Total free p.: 116 cont.: 2
biggest part: 116 code: 2
e> _

e> ps -la
PID PPID TIME WCHAN MEM CMD
0 0 432 r04 007 console
2 1 0 0 004 ps
1 0 0 r04 005 sh0
e> _

F1-NEXT SESSION F3-ADD A SESSION
```

İşte Linux ekranı...

C64'lü Albüm Kapağı

Emektar C64, insana “yuh be!” dedirtecek kadar ilginç işlere konu olabiliyor. İşte, 64'ü son olarak Norveçli goth/industrial grubu Apoptygma Berzerk'in Kathy's Song adlı single albümünün kapağında görüyoruz! Almak isteyenler [Amazon](https://www.amazon.com)'a bir göz atсын...



KISA KISA...

>> C=1'in kullandığı CPU olan 65816'nın programlama kitabı yeniden basılıyor. İlgilenenler (sepsypa@hotmail.com) adresine “For Mertin” başlıklı bir yazı yazabilirler...

>> Windows altında .d64 dosyalarını editlemeyi sağlayan "D64 Editor" (<http://almighty.c64.org/d64editor.html>) adresinde. Her türlü bilgi ve ekran görüntüleri aynı adreste bulunabilir.

>> Commodore Scene dergisinin 38'inci sayısı çıktı! Siparişler veya abonelik için: (<http://www.commodorescene.org.uk/>)

>> Almanca bir C64 dergisi : PDF formatında **LOTEK64**. Şimdilik 4 sayı. (<http://www.c64-mags.de/files.php?cat=33>)

>> Commodore Currents dergisi internet üzerinden yayın hayatına başladı. Derginin .pdf ve Postscript II sürümleri bulunuyor. (<http://www.geocities.com/eyethian2000/ccurrents.htm>)

>> Olağanüstü Commodore dergisi C=Hacking'in 1992'den bu yana çıkan 21 sayısını (<http://www.ffd2.com/fridge/chacking/>) adresinde bulabilirsiniz. Sayılar .txt formatında. Resimler ise ASCII art ile yapılıyor.

- >> Bir başka Commodore dergisi olan 3 sayılık Discovery için :
(<http://www.ffd2.com/fridge/discovery/>) (NOT : .GZ uzantılı olan dosyayı UNZIP yaptıktan sonra oluşan dosyaya .doc uzantısı ekleyin.)
- >> Computer Bild dergisindeki C=1 haberini görmek için
(http://www.c64.sk/files/C1_ComputerBild.jpg) adresine. (Almanca)
- >> Ünlü Commodore emülatörü VICE'in 1.11'inci sürümü çıktı. (<http://viceteam.bei.t-online.de/>)
- >> Commodore için teknik servis yazıları: (<http://personalpages.tds.net/~rcarlsen/>) Buradaki bilgilerle 64'ünüzün hemen her sorununu kendi başınıza halledebilirsiniz.
- >> Tiger disk: (<http://www.c64-mags.de/download.php?id=778>)
- >> Dekadence Xmas demo: (<http://www.dekadence64.org/xmas2002.zip>)
- >> Creators internet sitesi: (<http://mermaid.c64scene.org/creators/>) Farklı platformlar için (C64, VIC20, Atari ST vs.) birçok materyal bulunuyor.
- >> Covert BitOps internet sitesi: (<http://covertbitops.c64.org/>) C64 için oyun, müzik ve küçük utility programları yazan bir grup.
- >> Arts of Darkness grubunun internet sitesi: (<http://www.artsofdarkness.de/>)
- >> Türkiye'de çıkan Commodore/Amiga yayınları için bir site hazırlıyorum. Eğer sizde bende olmayan dergi ya da kitaplar varsa scan edip yollayabilirsiniz. Adres:
(<http://www.students.itu.edu.tr/~celikdeni>)
- >> Bu haber C64 ile ilgili değil ama yeri gelmişken Amigart'ın ve Türkiye Amiga Kullanıcıları Grubu'nun internet sitelerini de yazmak istedim. Adresler: (<http://www.amigart.com>) ve
(<http://www.amigart.com/tr>).
- Gruplar:
- >> <http://groups.google.com/groups?hl=en&lr=&ie=UTF-8&group=comp.sys.cbm>
- CommodoreOne Ana Sayfası
- >> (<http://c64upgra.de/c-one/>)

COMMODORE-ONE

Kadın-erkek, dede-torun demeden dört gözle beklediğiniz CommodoreOne rüyası en sonunda gerçek oluyor. Hani oldu da yanlışlıkla makinenin özelliklerini unuttuysanız diye bir hatırlatma yapalım dedik...

NASIL BİRŞEY?: CommodoreOne, micro-ATX tipi kasalara uyumlu olarak satılan bir anakarttan oluşuyor.

İŞLEMCİ: İşlemci olarak, 6502'ye tamamen uyumlu olan 65c816 kullanılıyor. 65c816 yaklaşık 20 MHz hızında çalışabiliyor ve 24 bit adres genişliğine sahip.

GÖRÜNTÜ: Yeni görüntü çipimizin adı **SuperVIC**... Monitör çıkışı olarak standart VGA çıkışı kullanılıyor. 65535 rengin 256 tanesi aynı anda kullanılabilir. Overscan özelliği var. 1280*1024 çözünürlüğe kadar çıkılabilir. Tüm video modlarında VIC-II uyumu var. İster klasik VIC, ister SuperVIC modunda kullanılabilir.

SES: C=1'de ses çipi olarak **MonsterSID** kullanılıyor. Bu, istenirse klasik SID emülasyonu da gerçekleştirebilen bir çip. MonsterSID modunda ise 16 değişik ses stereo olarak üretilebilir.

INTERNAL I/O: 1581 emülasyonu sağlayan 3,5" lik disket sürücü yuvası bulunuyor. 720'lik, 1.44'lük veya 2.88'lik sürücüler destekleniyor. IDE arayüzlü DMA desteği, Compact Flash Media yuvası ve (TTL LCD stili) LCD arayüzü de bulunmakta.

KART YUVALARI: Anakartın üzerinde 3 adet mini Zorro yuvası bulunuyor ki bunlar klasik Commodore kartuşlarıyla uyumluluk sağlayabiliyor. Ayrıca bir adet PCI yuvası da mevcut.

DİĞER İVİR ZIVIRLAR: C64 matrix emülasyonuna (ne?) sahip PS/2 klavye girişi ve 1351'e (C64'ün faresi) uyumlu PS/2 fare girişi de var. Ayrıca VIC, 64, 128 ve 264 sürücülerine ve yazıcılarına uyumlu IEC seri konektörü, bunun yanında iki joystick portu (keyboard uyumlu), yüksek hızlı RS-232 seri portu, paralel port ve ne olduğunu bilmediğim "Geek port" da eksik bırakılmamış...

OLASI SORULAR:

-CommodoreOne (C=1) C64 veya 128 için yeni bir kartuş mu?

>>Hayır, C=1 bir kartuş falan değil, tamamen yeni bir bilgisayardır...

-C=1'i çalıştırmak için nasıl bir PC'ye sahip olmalıyım? Ya da bir PC'im olmalı mı?

>>C=1 PC için tasarlanmış bir kart olmadığı gibi, bir emülatör de değildir. Bu yüzden C=1'e sahip olmak için bir PC sahibi olmak gibi birşey söz konusu değil. Dedik ya, yepyeni bir bilgisayardır...

-Eski programlarım C=1'de çalışır mı?

>>C=1, 65816 işlemcisini kullandığından C64 programlarının çok büyük bir çoğunluğu C=1'de de çalışacaktır. Hatta bazı programlar 65816 uyumluluğu için yamanmış durumda. Daha fazla bilgi için come.to/supercpu adresine bir gözatın..

-C=1'in bir teyp girişi var mı?

>>Şimdilik hayır, ama bunu sağlayacak bir arayüz yapılabilmesi mümkün. Mesela "Geek port" üzerinden...

-Commodore veya CMD sürücülerini ekleme yeteneği olacak mı?

>>Birçok Commodore ve CMD yan birimi ve çeşitli fast loader'lar test edilmiş. İyi çalıştığı söyleniyor.

-Peki, RS 232 portu bir kullanıcı portu bağlantısı mı olacak, yoksa gerçek bir COM portu mu? Gerçek bir 6551 ACIA'ya sahip olacak mı?

>>RS-232 seri portu, 2 adet Amiga 1200 uyumlu clock port konektörüyle değiştirilecek. Çünkü Amiga için olan Silversurfer seri portu tam bunun için. Yüksek sayılarda üretildiği için, yeni 16c550 UART'ları kullanmaktan daha ucuza geliyor. Ek olarak, bu porta uyumlu Amiga birimlerini kullanma fırsatı veriyor.

-İkinci bir işlemci olarak 6510 kullanılacağı doğru mu?

>>C=1'de isteyenin istediği CPU kartı takabileceği bir port bulunuyor (65816, 6502, 65C02, Z80, 6809 vs.). Böylece CPU kartını çıkarıp CompactFlash kartındaki programı değiştirerek, C=1'i 80'li yıllardaki herhangi bir bilgisayara çevirmek mümkün.

MonsterSID

-MonsterSID'in kaç tane filter'ı olacak?

>>Sağ ve sol olmak üzere iki.

-MonsterSID nasıl bir frekans aralığı kullanıyor?

>>MonsterSID'in frekans aralığı orjinal SID ile aynı.

-Aynı anda 8 DMA ve 16 SID sesini kullanabilecek miyiz?

>>Evet.

-Dijitize sesleri dinleyebilecek miyiz?

>>Evet.

-MonsterSID'de "line in" veya mikrofon girişi olacak mı?

>>Şu anda böyle bir plan yok.

SuperVIC

-C=1 ile nasıl bir monitör kullanmalıyım?

>>C=1, IBM-PC uyumlu bir VGA/SVGA monitörlerle çalışacak. Yüksek çözünürlükte daha fazla ekran modunu desteklediği için VGA seçildi. C=1 ile 1702 veya 1902 gibi eski Commodore monitörlerinizi veya televizyonunuzu

kullanamazsınız, çünkü C=1'in VGA çıkışı bu birimlerle uyumlu değildir (Aşağıdaki tabloya bakın). Ancak, (berbat bir görüntü kalitesi olan) VGA/TV çeviricilerini kullanırsanız çalışabilir ama haliyle bu da tavsiye edilmiyor.
-Peki o zaman niye C=1'in içine VGA/TV dönüştürücü eklenmiyor?

>>Bu çok pahalıya patlardı da ondan...

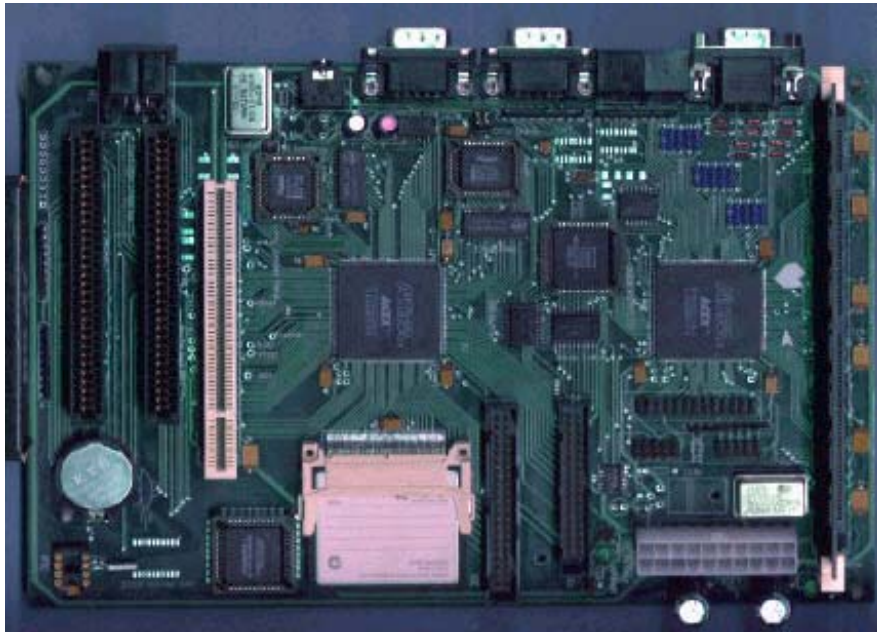
-C=1 ile birden fazla monitör kullanabilecek miyim?

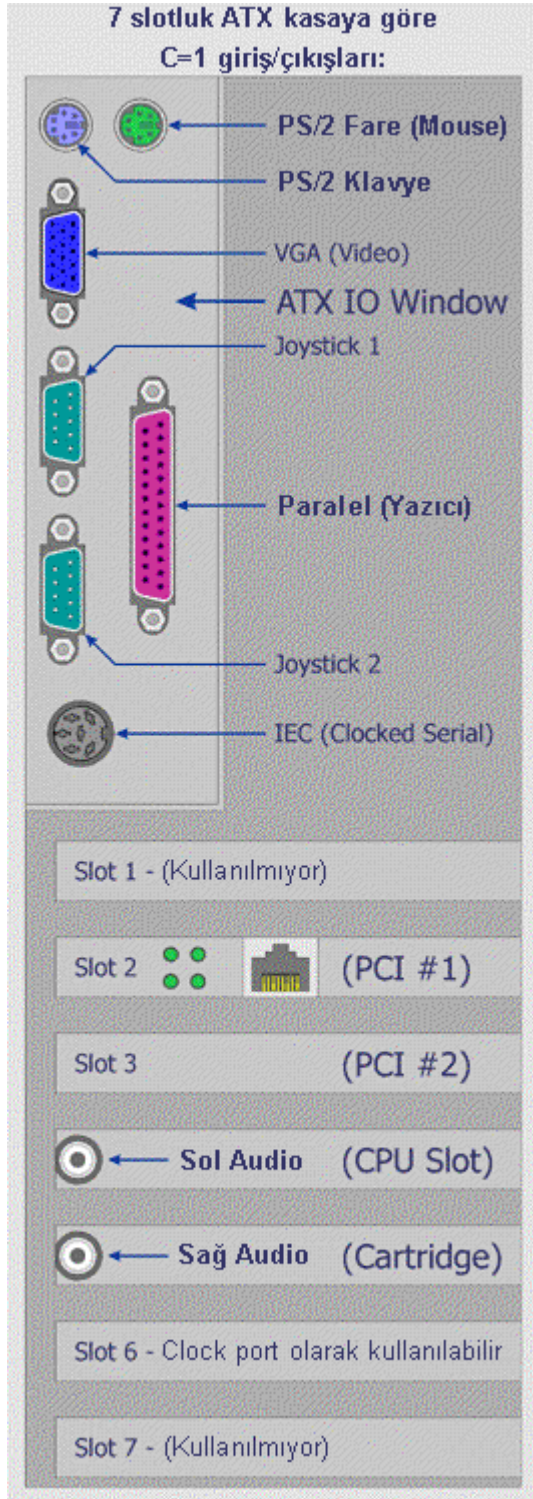
>>Şu anda bunun için bir plan yok.

Bu muhteşem makinenin fiyatının 249 Euro olması düşünülüyor. Daha fazla bilgi isteyenler [C=1 ana sayfasını](#) inceleyebilir...

Aşağıda C=1 ile çalışabilecek Commodore monitör modelleri ve bunların özellikleri yazılı. Yalnız belirtmekte fayda var, kimse bundan emin değil. Bu sadece bir varsayım:

1403	13 inç / VGA
1407	14 inç / VGA / Monochrome / 64 gri ton
1428	14 inç / Multi-sync / VGA
1450	Monochrome / BISYNC
1930	14 inç / VGA / .31mm dot pitch
1934	14 inç / VGA / .39mm dot pitch
1935-II	14 inç / SVGA / .28mm dot pitch / MPR-II düşük radyasyonlu
1936	14 inç / SVGA / .28mm dot pitch
1940	13 inç / Amiga Multiscan Monitör / .39mm dot pitch / Hoparlör
1942	13 inç / Amiga Multiscan Monitör / .28mm dot pitch / Hoparlör
1950	14 inç / MultiScan
1960	14 inç / MultiScan





C=1 için tasarlanan klavye:

*	Ex.0-4	Pause Break
---	--------	-------------

Ex.0-0	Ex.0-1	Ex.0-2	Ex.0-3
--------	--------	--------	--------

F6	F6	F8	F8
F5	F6	F7	F8

F2	F2	F4	F4
F1	F2	F3	F4

RUN STOP

INST	CLR HOME	£	π ↑
DEL	@		

INST	DEL	↑	Shift-M
------	-----	---	---------

Q	W	E	R	T	Y	U	I	O	P	[]	Shift-M
---	---	---	---	---	---	---	---	---	---	---	---	---------

A	S	D	F	G	H	J	K	L	;	'	~	8	9	0)	=	+	INST	DEL	↑	Shift-M
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	------	-----	---	---------

CONTROL	L.SHIFT	L.SHIFT	>	<	Z	X	C	V	B	N	M	,	<	.	>	/	?	R. SHIFT	CONTROL
---------	---------	---------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----------	---------

Crsr Up	Crsr Down	Crsr Left	Crsr Right	(S-Crsr Left)	(S-Crsr Right)
---------	-----------	-----------	------------	---------------	----------------

Ex.1-5	HOME	7	8	9	*	+	RETURN
--------	------	---	---	---	---	---	--------

Ex.1-5	HOME	7	8	9	*	+	RETURN
--------	------	---	---	---	---	---	--------

Ex.1-5	HOME	7	8	9	*	+	RETURN
--------	------	---	---	---	---	---	--------

Ex.1-5	HOME	7	8	9	*	+	RETURN
--------	------	---	---	---	---	---	--------

ASSEMBLER KURSU 2

Bu sayıdan itibaren 6510 işlemcisinin registerlerini, adresleme modlarını, şartlı dallanma ve alt program nedir gibi konuları göreceğiz.

REGİSTERLER

Makine dili komutları her biri basit aritmetiksel, mantıksal, taşıma veya şartlı dallanma işlemi yapan bir baytlık kodlardan oluşur. Bu kodu komutun çeşidine göre bir başka kod veya o koda ait 1 veya 2 baytlık bir parametre (sabit sayı veya adres) izler.

Komutlara başlamadan önce 6510'un registerlerini bilmemiz gerekmektedir. Çünkü registerleri bilmeden komutlara tam anlamı ile hakim olamazsınız. Peki register ne demektir? Register kelime olarak karşılığı kaydedici demektir. Mikroişlemcilerde ise registerler mikroişlemci içinde bulunan işlem yapan birimlerdir. Makina dili komutları aslında çok ilkeldir. Çeşitli programlama dillerindeki gibi değişkenler, karmaşık işlemleri kolayca gerçekleştiren aritmetiksel işlemler veya INPUT, PRINT vs. komutlar yoktur. Ekrana basit bir "A" harfi yazmak için bile hangi registerlerin/adreslerin kullanıldığını, karakterlerin ASCII kodlarını ve birde C64'e özel çeşitli renk, kursör kodlarını vs bilmeniz gerekmektedir. Bu nedenle C64'ün bellek haritası ile ASCII kodları elinizin altında olmalıdır. Bir makina dili komutu her seferinde sadece tek bir iş yapar. Komutların başlıca görevlerini aşağıdaki gibi sınıflandırabiliriz.

1. Bir bellek adresinin değerini değiştirmek (arttırmak, azaltmak vs) veya bir adresin değerini başka bir adrese kopyalamak.
2. Bir baytlık bir sayı üzerinde basit aritmetik veya mantık işlemleri yapmak.
3. Program akışı içinde şartlı veya doğrudan başka bir adrese dallanmak.

Bütün bu işlemleri 6510'un registerleri yapar. Registerleri bir çeşit taşıyıcı olarak düşünebilirsiniz. 6510'un registerleri A, X, Y, ST, PC, SP, ve I/O port'udur.

Şimdi bunları sırayla görelim:

1. A (Akümülatör) : Mikroişlemcinin en önemli registeridir. Çeşitli komutlar kullanarak aküdeki bilgiyi başka bir adrese, registre veya yığına atabilir, buralardaki bilgiyi aküye yükleyebilirsiniz. Akünün en büyük özelliği aritmetiksel ve mantıksal işlemleri yapabilen tek register olmasıdır.

2. X (X index registeri) : X registerindeki bilgiyi aküye, herhangi bir adrese veya SP'ye atabilir, buralardaki bilgiyi X'e yükleyebilirsiniz. X registerini kullanarak döngü kurabilirsiniz. Ayrıca sadece X registeri kullanılarak yapılan işlemler de vardır.

3. Y (Y index registeri) : Y registerindeki bilgiyi aküye, herhangi bir adrese atabilir, buralardaki bilgiyi Y'ye yükleyebilirsiniz. Y registerini kullanarak döngü kurabilirsiniz. Ayrıca sadece Y registeri kullanılarak yapılan işlemler de vardır.

4. ST (STATUS -durum- registeri) : Bu registerin diğer registerlerden oldukça farklı bir işlevi vardır. ST registerini oluşturan 8 bitin her birinin bir görevi vardır ve mikroişlemcinin o anki işlediği komuta göre çeşitli durumları evet/hayır gibi gösteren birer bayrak (flag) olarak kullanılır. Statü registerinin bitleri "0" veya "1" olmalarına göre aşağıdaki durumlarla ilgili bilgileri verir.

BİT NO	7	6	5	4	3	2	1	0
FLAGLAR	N	V	-	B	D	I	Z	C

BİT	AÇIKLAMA
7	N : (Negatif) Bir önceki işlemin sonucu negatifmi ?
6	V : (Overflow) Bir önceki işlemin sonucu 127'den büyükmü ?
5	Boş. Herzaman "1" dir.
4	B : (Break) Program kesintiye uğramış mı ?
3	D : (Decimal) İşlemci desimal (onlu) işlem modundamı ?
2	I : (Interrupt) İnterrupt durdurulmuş mu ?
1	Z : (Zero) Bir önceki işlemin sonucu "0" mı ?
0	C : (Carry) Bir önceki toplama, çıkarma veya karşılaştırma işleminde elde varmı ?

5. PC (Program Counter -Program Sayacı-) : Program Sayacı işlemcinin o anda işlenmekte olan makina dili komutunun bulunduğu adresi gösterir ve sürekli değişir. Nedeni ise C64'ün işletim sisteminin sürekli çalışır durumda olmasıdır. PC'yi durdurmak için işlemcinin durdurulması gerekir. Mesela hatalı bir işlem yapıldığında C64 kilitlenirse işlemci durmuş demektir. Bu durumda işlemcinin resetlenmesi gerekir.

6. SP (Stack Pointer -Yığın Göstergesi-) : Bu register, yığındaki ilk boş yerleşim yerinin adresini gösterir. Yığın (Stack) makina dili ile yazılmış programlarda geçici saklama yeri olarak kullanılır. Ayrıca ana programın ve JSR komutu ile çağrılan alt rutinlerden RTS komutu karşılığınca geri döneceği adresi gösterir. Yığın ile işlem yapılırken dikkatli olun. Çünkü eksik bir komut bilgisayarın kilitlenmesine veya programın rastgele bir adresten itibaren çalışmasına sebep olabilir. Eksik işlem'den kastettiğim kullanılan her PHA komutuna karşılık bir PLA komutu kullanılmalıdır. Yığın LIFO (Last In First Out) prensibiyle çalışır. Yani yığma en son attığınız bir sayı yığından ilk önce alınır. Yığma bir sayı attığınızda, SP 1 azalır. Sayıyı yığından geri aldığınızda, SP 1 artar. Yığın dediğimiz bölge, C64'ün RAM'inde 256 ile 512 nolu adresler arasındaki bölgedir (512 hariç). Yığın göstergesi bu bölgedeki ilk boş yeri gösterir.

7. I/O PORT (Giriş/Çıkış Portu) : C64'ün hafızasındaki 0 ve 1 nolu adresler işlemcinin kendi içindedir. 0 nolu adres veri yön registeri olup 1 nolu adresteki verinin yönünü belirler. 1 nolu adres ise veri registeridir. Bu iki adres C64'ün hafıza düzenleme işleminde kullanılır. Böylece C64'ün bellek haritasındaki RAM-ROM-I/O bölgesi gibi birimler seçilebilir. Bu iki adresi bu sayıdaki **"bellek haritası"** bölümünde ayrıntılı olarak göreceğiz.

MAKİNA DİLİNDE NASIL PROGRAM YAZABİLİRİZ

Makina dili programlarını hafızaya yüklediğinizde bu programları düzeltme imkanınız yoktur. Bu nedenle makina dili programları yazmak için genellikle ASSEMBLER EDİTÖR denilen programlar kullanılır. (Benim sürekli kullandığım TURBOASSEMBLER programıdır.)

Bu programlar makina dili komutlarının anlaşılması için, bir sürü sayı ile uğraşılması yerine, anlaşılması ve akılda kalması daha kolay "MNEMONIC" denilen kısaltılmış komut isimleri kullanmamızı sağlar. Kısaca makina dili programlarını "MNEMONIC" kullanarak yazmamızı sağlayan programlara "ASSEMBLER", bu işlemin tersini yapan, yani hafızadaki bir makina dili programını "MNEMONIC" şeklinde gösteren programlara "DISASSEMBLER" denir.

Bu iki programdan başka içinde basit bir assembler/disassembler programı içeren ve çeşitli işlemler yapılmasını sağlayan - mesela hafızanın bir bölümünü save yapan veya belli bir sayıyla dolduran veya başka bir hafıza bölümüyle karşılaştıran vs. - yardımcı programlardan oluşan "UTILITY" programlarına ise "MONİTOR" programı denir.

ADRESLEME MODLARI

6510 işlemcisi bize 8 çeşit adresleme modu sunmaktadır. Bazı komutlar bu modların hepsini kullanır, bazı komutlar ise daha az adresleme moduna sahiptir. Adresleme modları yazdığımız programların daha kısa olmasını sağlar.

Dolaysız Adresleme : Bu adresleme modunda A, X ve Y registerlerine doğrudan bir değer yüklenir. Bu adresleme modunu kullanırken # \$ işaretlerini kullanacağız.

Örnek : LDA #\$00 LDX #\$3A LDY #\$FF

Bu adresleme modundaki komutlar 2 bayt uzunluğundadır.

Sıfırıncı Sayfa Adresleme : Bu mod, bellek haritasında 0 ile 255 (dahil) arasındaki adresler ile işlem yaparken kullanılır. Dolaysız adresleme modu haricindeki adresleme modları için \$ işaretini kullanacağız.

Örnek : LDA \$02 ASL \$FB STX \$01

Bu adresleme modundaki komutlar 2 bayt uzunluğundadır.

Mutlak Adresleme : Mutlak Adresleme modunu Sıfırıncı Sayfa Adresleme modunun genişletilmiş olarak düşünün. Yani Sıfırıncı Sayfa Adresleme modu aslında Mutlak Adresleme Modunun yüksek adresi gösteren baytının her zaman 0 olan halidir.

Örnek : INC \$DEDE ROR \$CEFA ADC \$BABA

Bu adresleme modundaki komutlar 3 bayt uzunluğundadır.

Mutlak X İndeksli Adresleme : Bu modda belli bir taban adresine X registeri eklenerek bulunan yeni adresle işlem yapılır. Mesela taban adres \$2000 olsun X registerinde ise \$30 (desimal 48) bulunsun, işlem yaparken $\$2000 + \$30 = \$2030$ adresi kullanılır.

Örnek : LDA \$3A00,X SBC \$BACA,X DEC \$EBEE,X

Bu adresleme modundaki komutlar 3 bayt uzunluğundadır.

Mutlak Y İndeksli Adresleme : Yukarıdaki modun Y registerini kullanan şekli olarak tanımlayabiliriz. Mutlak X indeksli ve mutlak Y indeksli adresleme modları özellikle hafızada büyük veri transferleri sırasında ve döngü içinde adreslere sabit değer yüklerken kullanabilirsiniz.

Sıfıncı Sayfa X İndeksli Adresleme : Bu mod Sıfıncı Sayfa Adresleme moduna X registeri eklenerek kullanılır. Mutlak X indeksli adresleme modunun kısaltılmışı olarak düşünebilirsiniz.

Dolaylı İndeksli Adresleme : İşte geldik kullanımı biraz karışık olan bir adresleme moduna... Aslında çalışma mantığını bilerseniz çok rahat kullanırsınız. Bu mod sıfıncı sayfada 2 ardışık (arka arkaya) adresle ve sadece Y registeriyle kullanılır. Bu moddaki komutların yazılışı

KOMUT (\$adres),Y şeklindedir.

Çalışma şekli şöyledir : İşlem yapacağınız bir taban adresini LOW BAYT-HIGH BAYT formatında (bkz: assembler-1) Sıfıncı Sayfada arka arkaya gelen 2 adrese yerleştirin. Y registerine ise bir değer verin. Komut bu iki adreste bulunan değerlerin gösterdiği adrese Y' nin değerinin eklenmesiyle bulunan yeni adresi kullanarak işlem yapar. Komutunuz mesela LDA (\$02),Y olsun. Daha önceden ise \$C5A3 sayısını \$A3 ve \$C5 olarak \$02 ve \$03 adreslerine yükleyelim. Y registerinde de \$12 (desimal 18) bulunsun. Bu durumda bizim komutumuz \$02 ve \$03 adreslerindeki değerlerin gösterdiği \$C5A3 sayısına Y' deki \$12 değerini ekleyerek bulunan \$C5B5 adresindeki değeri aküye yükler. Yani aslında verdiğim örnek LDA \$C5A3,Y komutunun değişik bir şeklidir. Aradaki tek fark LDA (\$02),Y komutunun 2 bayt LDA \$C5A3,Y komutunun 3 bayt sürmesidir.

Bu arada Dolaylı İndeksli komutları kullanırken Sıfıncı Sayfada rastgele iki adresi kullanamazsınız. Çünkü işletim sistemi bu bölgeyi kendisi için kullanır ve bize kalan sadece **\$FB, \$FC, \$FD ve \$FE** adresleridir. Ben yazdığım programlarda \$02 ve \$03 adreslerini de kullanırım.

İndeksli Dolaylı Adresleme : En sona kalan bu mod ne yazık ki kullanımı oldukça dikkat isteyen ve benim şu ana kadar hiç kullanmadığım bir adresleme modudur. Üstteki mod sadece Y registeriyle kullanıldığı gibi bu modda sadece X registeriyle kullanılır.

Çalışma şeklini şöyle açıklayalım : Sıfıncı Sayfada bir bayt olarak belirtilen adresten X ötedeki adreste ve onu izleyen adreste bulunan değerlerin gösterdiği adres işlem yapılacak yeni adrestir.

Örnek : Taban adresimiz \$20 olsun. X registerinde \$50, \$70 nolu adreste \$A5, \$71 nolu adreste ise \$85 olduğunu kabul edelim. Komutumuz ise **LDA (\$20,X)** olsun. Bu durumda taban adres olan \$20 sayısına X' in değeri olan \$50 eklenir. Sonuç \$70 tir. Daha sonra \$70 (LOW BAYT) ve \$71 (HIGH BAYT) nolu adreslerdeki değerlerin oluşturduğu sayı olan \$85A5 nolu adresteki sayı Aküye yüklenir.

ŞARTLI DALLANMALAR

Önce şartlı dallanmanın ne demek olduğunu açıklayalım. Program içinde yapılan bir aritmetiksel, mantıksal, arttırma, azaltma, karşılaştırma veya yükleme işleminde sonuca göre başka bir adrese dallanıp yani sıçrayıp programın akışını değiştirebilirsiniz. Daha genel bir ifadeyle programda o anda işlenmekte olan komut sonucunda eğer **N, Z, C ve V** flaglarından değişenler olursa, bu değişimlere göre program içinde sıçrama yapabilirsiniz.

Şunu unutmayın: Flaglara göre yapılan sıçramalarda gideceğiniz yer çok uzakta olmamalıdır.

Yazdığımız programlarda eğer uzak bir mesafeye sıçramaya çalışırsanız kullandığınız editör programı hata verecek ve hatanın hangi satır(lar)da olduğunu yazacaktır. Şartlı dallanmalarda, dallanma en fazla 127 bayt ileriye veya en fazla 128 bayt geriye olabilir.

ALT PROGRAMLAR

Alt program nedir? Önce tanımını yapalım. Bir program içinde farklı yerlerden çağrılan ve sadece belli işleri yapan ufak programcıklara alt program denir. Bir program yazmaya başladınız ve bazı yerlerde aynı işlemi yapan bölümler oluşmaya başladı. Bu durumda aynı işlemi yapan bölümü ayrı bir program olarak yazarsınız. Ana program içinde gerekli yerlerde **JSR** komutunu kullanarak alt programa sıçarsınız, gerekli işlemler yapıldıktan sonra alt programın sonundaki **RTS** komutuyla ana programdaki alt programı çağırın komuttan bir sonraki komut işleme konulur ve ana program kaldığı

yerden devam eder. Alt programları öyle yazmalısınız ki yazacağınız programlarda hazır halde olsun. Mesela bir intro yapmak ve bu intro da renk şeritleri olmasını istiyorsunuz. Önce renk şeritleri rutinini yazarsınız ve ana programda ilgili raster satır değerini vererek renk şeritleri rutinini çağırırsınız. Commodore 64'ün ROM'larında işinize yarayan birçok alt program mevcuttur. (İsterseniz alt rutin diyebilirsiniz. Bundan sonra hep alt rutin kelimelerini kullanacağız.) İşte birkaç hazır alt rutin adresi.....

\$E544 ----- Ekranı siler.

\$FFD2 ----- Ekrana bir karakter yazar.

\$FFE4 ----- Klavyede basılan tuşu okur.

Gördüğünüz gibi bazı hazır alt rutinler var. Bazı alt rutinler giriş parametreleri ister, bazıları ise yapılan işlemin sonucunu bir registerde veya adreste tutar. Hazır başlamışken adresleri inceleyelim.

\$E544 : Aslında ekran silme alt rutini C64'ün ilk açılışta yerine getirdiği -INITIALIZE VIDEO CONTROLLER- işleminin bir bölümü. \$E518 adresinden başlayan bu rutinde kursör yanıp sönme, zamanlama vs. gibi işlemlerinden başka yazı rengini belirleyen \$0286 adresine #\$0E değeri verilerek yazının açık mavi olması sağlanmaktadır. Eğer yazdığınız programlarda başlangıçta yazı rengini de değiştiriyorsanız mesela **LDA #\$01 - STA \$0286 - JSR \$E544** gibi komutlarınız varsa bu işi **LDA #\$01 - JSR \$E536** yazarak ta yapabilirsiniz. Böylece programınız 3 bayt kısalır. Belki önemli bir kısalma değildir ama PC'lerdeki gibi MegaBayt ile ifade edilen büyüklükte RAM'ımız olmadığı için RAM'ı iyi kullanmak lazım ☺☺. Peki istediğiniz gibi hazır alt rutinler yoksa ne yapacaksınız ? Diyelimki ekranın tamamını değilde ortadan 10 satır sileceksiniz. İki çözüm var. Ya ilgili ekran satırlarının adreslerini tek tek hesaplayıp ekrana BOŞLUK karakteri basacaksınız veya yine KERNAL ROM'da bulacağınız bir alt rutinle işinizi halledeceksiniz. Bu alt rutin, ekranı silen alt rutin tarafından kullanılan ve **\$E9FF** adresinde bulunan **CLEAR SCREEN-LINE** isimli başka bir alt rutindir.

\$FFD2 adresi ise aküde tutulan baytı o anki çıkış cihazına -Normalde ekrandır- gönderir. Mesela **LDA #\$41 - JSR \$FFD2** komutları ile ekrana "A" harfini basarsınız.

Son olarak **\$FFE4** adresini görelim. Bu adres ile o anki giriş cihazından -Normalde klavyedir- bir bayt okur ve sonucu aküde tutar. Mesela **JSR \$FFE4 - CMP #\$41** komutları ile klavyede "A" tuşuna basılıp basılmadığını kontrol edebilirsiniz. Gördüğünüz gibi \$E544 parametresiz kullanılır. \$FFD2 aküyle beslenir ☺☺ \$FFE4 ise Aküyü besler. (Akü şarj cihazı yani ☺☺) Ana programınızda, alt rutinlerinizi ana programınızın bütünlüğünü bozmayacak şekilde istediğiniz yere yerleştirebilirsiniz ve ana programınızın çalışmasını etkilemeyecek herhangi bir yerden alt rutinlerinizi istediğiniz kadar çağırabilirsiniz. Sadece mutlak adreslemeli komutlarda dikkatli olun yeter.

Commodore 64'ün ROM'larındaki rutinleri, daha doğrusu romların içinde ne var ne yok öğrenmek istiyorsanız **ANATOMY OF THE COMMODORE 64** isimli kitabı bulmalısınız. Kitabın orijinalini bulamazsanız bile internette TXT ve HTML uzantılı dosyalar halinde bulmanız mümkün. Yeter ki arayın. (Deniz'in notu: Millet, gördüğünüz gibi bu adamdan hayır yok. Neyse ben kitabın yazarlarını ve basımevini yazıyorum: Angerhausen/Becker/English/Gerits - Abacus Software - Data Becker GmbH. Kitap [Amazon](#)'da ikinci el olarak bulunuyor. [Lemon64](#)'te de kapağının resmi var. Ayrıca [Centsible Software](#)'de, [Commodore Computer Center](#)'da, [Dr. Feelgood's Crazy C= Sales](#)'de (evet, kötü isim) ve [Vintage Computer](#)'da hâlâ bazı kitaplar (ikinci el veya yeni olarak) mevcut.)

PROGRAM KÖŞESİ

0	1	2	3
4	5	6	7
8	9	A	B
C	D	E	F

\$A474=42100

HEX - DEC CEVİRİCİ
KASIM/2002 (C)HADES

HEX – DEC ÇEVİRİCİ

Assembler kursunun ilk bölümünde gördüğümüz HEX-DEC çevirme işlemini bırakalım C64 yapsın. HEX – DEC CONVERTER isimli programımızı çalıştırınca HEX sayı sisteminde kullanılan tuşlar bulunan bir görüntü karşınıza gelecektir. Tuşlara bastıkça ekrandaki ilgili karakter flaş yapacak ve sağ tarafta yazılacaktır. Dört kez basınca yazmış olduğunuz HEX sayının onaltılı karşılığı yazılacaktır. SPACE tuşuna basılırsa yeni sayıyı girebilirsiniz. Bir baytlık sayı girmek için – mesela \$FA olsun – sayı 00FA olarak girilmelidir.

```
;-----;
; hex-dec cevirci ;
; (c) 11/2002  hades;
;-----;

*=  $0900          ;Programın başlangıç adresi

lda #$0f          ;Açık Gri renk kodu
sta $d020         ;Çerçeve rengi olsun
sta $d021         ;Ekran rengi olsun

lda #$0b          ;Koyu Gri renk kodu yazı rengi
jsr $e536         ;olsun ve ekran temizlensin.

lda #<text        ;Yazının alçak baytını yükle
ldy #>text        ;Yazının yüksek baytını yükle
jsr $able         ;Yazıyı yazdır.

yeni             clc          ;"Elde" bitini sıfırla
                 ldx #$07     ;Kursör pozisyonu satır değeri
                 ldy #$09     ; " " " sütun "
                 jsr $fff0     ;Kursörü yerleştir

                 lda #$24     ; "$" işaretini
                 jsr $ffd2     ;Ekrana bas

                 lda #$00     ;Kursör
                 sta $cc      ;gözüksün ve flaş yapsın

                 jsr bayt     ;Klavyeden bir bayt gir
                 sta $fc      ;Yüksek bayt olarak sakla

                 jsr bayt     ;Klavyeden bir bayt gir
                 sta $fb      ;Alçak bayt olarak sakla
                 lda #$3d     ; "=" işaretini
                 jsr $ffd2     ;Ekrana bas
```



```

        lda $fc                ;Yüksek bayt aküye
        ldx $fb                ;Alçak bayt X registerine
        jsr $bdc              ;Onlu sayıya çevir ve ekrana bas

tekrar   jsr $ffe4             ;Tuşları kontrol et
        cmp #$20              ;"SPACE" mi ?
        bne tekrar           ;değilse bir daha dene

clear    ldx #$00              ;X registerini sıfırla
        sta $0552,x          ;Aküdeki değeri ekrana bas. Aküde "SPACE"
                                ;olduğu için ekrana boşluk konacaktır.
        inx                  ;X'i 1 arttır
        cpx #$0b              ;X=11 oldu mu ?
        bne clear            ;olmadıysa tekrar

        jmp yeni              ;olduysa yeni işlem için başa dön.
;-----
bayt     jsr hextus           ;HEX sistem tuşlarını kontrol et

        lda temp              ;Geçici değeri oku
        asl                   ;Bir kez sola kaydır
        asl                   ; " " " "
        asl                   ; " " " "
        asl                   ; " " " "
        sta $fb              ;Üst nibble olarak sakla

        jsr hextus           ;HEX sistem tuşlarını kontrol et

        lda temp              ;Geçici değeri oku
        ora $fb              ;Üst nibble ile "OR" işlemi yap
        rts                  ;Geri dön

hextus   jsr $ffe4             ;Tuşları kontrol et
        cmp #$30              ;"0" mı?
        bcc hextus           ;küçükse tekrar oku
        cmp #$3a              ;":" mı?
        bcs harf             ;eşit veya büyükse harf kontrolüne git
        pha                  ;basılan tuşun (ASCII) değerini yığına at
        sec                  ;Elde bitini "1" yap
        sbc #$30              ;Tuşun ASCII değerinden $30 (48) çıkart
        jmp devam            ;zıpla!

harf     cmp #$41              ;Basılan tuş "A" mı?
        bcc hextus           ;küçükse tuşları tekrar oku
        cmp #$47              ;"G" mi?
        bcs hextus           ;eşit veya büyükse tuşları tekrar oku
        pha                  ;Basılan tuşun (ASCII) değerini yığına at
        sec                  ;"Elde" bitini "1" yap
        sbc #$37              ;Tuşun ASCII değerinden $37 (55) çıkart

devam    sta temp              ;Sonucu geçici olarak sakla
        asl                   ;Sonucu 2 ile çarp
        tax                   ;X = A
        lda colmem,x          ;COLOR RAM için adresin LOW baytını al
        sta $fd               ;Sakla
        lda colmem+1,x        ;COLOR RAM için adresin HIGH baytını al
        sta $fe               ;Sakla

        ldy #$00              ;Y registerini sıfırla
        lda ($fd),y           ;Renk Ram'ındaki adresin değerini oku
        sta oldcol            ;Sakla

```

```

        lda newcol          ;Yeni renk deęerini al
        sta ($fd),y         ;Renk Ram'ındaki adrese yaz

        pla                 ;Yığındaki deęeri al
        jsr $ffd2           ;Ekрана bas

delay    ldy #$a0           ;Y registerine $A0 y kle (Gecikme sayacı)
        ldx #$00           ;X registerini sıfırla
        jsr $eeb3          ;ROM'daki 1 milisecond delay rutinine git
        dey                ;Sayacı 1 azalt
        bne delay          ;"0" olmadıysa gecikmeye devam

        lda oldcol         ;Eski renk deęerini oku
        sta ($fd),y        ;Renk Ram'ındaki adrese geri yaz
        rts                ;Alt rutinden geri d n
;-----
;Deęişkenler, tablo(lar), yazılar vs. bu kısımda bulunmaktadır.
;-----
oldcol    .byte 0
newcol    .byte 7
temp      .byte 0
colmem    .word $d829,$d82b,$d82d,$d82f
          .word $d879,$d87b,$d87d,$d87f
          .word $d8c9,$d8cb,$d8cd,$d8cf
          .word $d919,$d91b,$d91d,$d91f

text      .byte $8e,$d5,$c3,$b2,$c3,$b2
          .byte $c3,$b2,$c3,$c9,$0d,$c2
          .byte $30,$c2,$31,$c2,$32,$c2
          .byte $33,$c2,$0d,$ab,$c3,$db
          .byte $c3,$db,$c3,$db,$c3,$b3
          .byte $0d,$c2,$34,$c2,$35,$c2
          .byte $36,$c2,$37,$c2,$0d,$ab
          .byte $c3,$db,$c3,$db,$c3,$db
          .byte $c3,$b3,$0d,$c2,$38,$c2
          .byte $39,$c2,$41,$c2,$42,$c2
          .byte $0d,$ab,$c3,$db,$c3,$db
          .byte $c3,$db,$c3,$b3,$0d,$c2
          .byte $43,$c2,$44,$c2,$45,$c2
          .byte $46,$c2,$0d,$ab,$c3,$b1
          .byte $c3,$b1,$c3,$b1,$c3,$b1
          .byte $c3,$c3,$c3,$c3,$c3,$c3
          .byte $c3,$c3,$c3,$c3,$c3,$c9
          .byte $0d,$c2,$20,$48,$45,$58
          .byte $20,$2d,$20,$44,$45,$43
          .byte $20,$43,$45,$56,$49,$52
          .byte $49,$43,$49,$c2,$0d,$c2
          .byte $4b,$41,$53,$49,$4d,$2f
          .byte $32,$30,$30,$32,$20,$28
          .byte $43,$29,$48,$41,$44,$45
          .byte $53,$c2,$0d,$ca,$c3,$c3
          .byte $c3,$c3,$c3,$c3,$c3,$c3
          .byte $c3,$c3,$c3,$c3,$c3,$c3
          .byte $c3,$c3,$c3,$c3,$c3,$cb
          .byte $05,$0d,$00,$00,$00,$00

;programın sonu

```

BELLEK HARİTASI – 2

Bellek haritasına kaldığımız yerden devam ediyoruz. Aslında geçen sayıda \$0000 - \$FFFF adresleri arasını bitirmiştik. Fakat **0 ve 1** numaralı adreslerle **\$D000 - \$DFFF** arasında yer alan ve kısaca **I / O bölgesi** olarak tanınan bellek bölümünün ayrıntılı incelemesini bu sayıda yapıyoruz.

Commodore 64'te 64 Kbaytlık RAM, 8 Kbaytlık BASIC ROM, 8 Kbaytlık KERNAL ROM, 4 Kbaytlık KARAKTER ROM ve 4 Kbaytlık I / O (Giriş / Çıkış) bölgesi vardır ve bütün bunlar toplam 88 Kbaytlık bir yer kaplamaktadır. Bunlardan başka uygun donanım kullanılarak kartuş denilen parçalarında birer bellek olarak kullanıldığını hesaba katarsak sayı artacaktır. Mesela ülkemizde bilinen adıyla MULTI ICE 6 kartuşunda 32 K'lık bir EPROM ve 8 K'lık bir RAM vardır. Ayrıca 512 K'lık RAM genişletme kartuşlarında bulunmaktadır. Kartuşlardaki bellekleri kullanabilmek için C64'ün hardware yapısını bilmek gerekmektedir. Bu bölümde sadece C64'ün içindeki bellek bölümlerine nasıl erişileceğini göreceğiz.

C64'ün işlemcisi olan 6510 – genel amaçlı kullanılabilen 6502'nin özel bir versiyonudur – 16 bitlik adres hattıyla en fazla 64 K adresleyebilmektedir. Fakat yukarıdaki 88 K'lık bir alan nasıl oluyorda kullanılabiliyor?. Bu işin sırrı 6510'un içindedir. 6502'nin özel bir versiyonu olması dolayısıyla 6510 aynı zamanda bir MMU (Memory Managment Unit – Bellek Yönetim Birimi) olarak kullanılmaktadır. Normal şartlarda RAM'da yer alması gereken 0 ve 1 numaralı adresler işlemci tarafından kendi içindeymiş gibi kullanılır. Bu iki adresten 0 numaralı olanı veri yön portu olup 1 numaralı adresin hangi bitlerinin giriş veya çıkış olarak kullanılacağını düzenler. Şöyle bir örnek verelim. 1 numaralı adresin Bit'lerini kapı olarak düşünün. 0 numaralı adres bu kapıların hangilerinden çıkış, hangilerinden giriş yapılacağına karar verir. 1 numaralı adres ise bu kapıların açık veya kapalı olacağını belirler.

0 numaralı adres yani VERİ YÖN PORT'u işlemcinin içinde olup dışarıyla bağlantısı yoktur. 1 numaralı adres ise KONTROL PORT'udur ve 6 adet çıkış ile C64'ün içindeki diğer yerlere bağlıdır. VERİ YÖN PORT'unda 47 (\$2F) değeri bulunmaktadır ve bu ikili sistemde **%101111** 'dir. Burada "1" ler çıkış, "0" lar giriş olarak tanımlanmıştır. Buna göre 1 numaralı adresin 0, 1, 2, 3 ve 5 numaralı bitleri çıkış 4 numaralı biti ise giriş olarak kullanılmaktadır.

C64'te bellek düzenleme işleminde 1 numaralı adresin 0, 1 ve 2 numaralı bitleri kullanılır. Diğer bitler ise teyp ile yapılan işlemlerde kullanılır. Şimdi bunlar inceleyelim.

İSİM	BİT	YÖN	AÇIKLAMA
LORAM	0	ÇIKIŞ	\$A000 - \$BFFF (BASIC) arasındaki RAM / ROM için kontrol.
HIRAM	1	ÇIKIŞ	\$E000 - \$FFFF (KERNAL) arasındaki RAM / ROM için kontrol.
CHAREN	2	ÇIKIŞ	\$D000 - \$DFFF arasında yer alan Giriş / Çıkış yada ROM için kontrol.
	3	ÇIKIŞ	Kasete yazma hattı.
	4	GİRİŞ	Teyp anahtarı duyumu.
	5	ÇIKIŞ	Teyp motor kontrolü.

1 numaralı adresin normal açılış değeri \$37 (55) olup bizi ilgilendiren ilk üç bit için konuşursak, C64 ilk açıldığında BASIC ROM; KERNAL ROM ve GİRİŞ / ÇIKIŞ bölgesi seçilmiş durumdadır. Yani LORAM; HIRAM ve CHAREN bitleri ilk açılışta "1" dir.

Eğer LORAM bitini "0" yaparsanız artık BASIC ROM devreden çıkar ve yerine aynı adreslerdeki RAM kullanılabilir hale gelir. BASIC programlama ile ilgilenmiyorsanız ve daha çok assembler kullanıyorsanız 8 K'lık bir bölge kazancınız olacaktır.

Eğer HIRAM bitini "0" yaparsanız anında bilgisayarın kilitlendiğini göreceksiniz. Çünkü KERNAL ROM'da bulunan ve C64'ün çalışmasını sağlayan rutinler devre dışı kalacak ve işlemci o rutinlerin yer aldığı adreslerde RAM boş olduğu için ne iş yapacağını bilemeyecektir. Bu nedenle eğer HIRAM'ı "0" yaparsanız – ki ancak makine diliyle olacağı için- değişiklikten önce SEI komutuyla interrupt'ları durdurmanız gerekmektedir.

Son olarak CHAREN bitini "0" yaparsanız artık I / O çiplerine –görüntü,ses vb.- erişemez fakat KARAKTER ROM'unu okuyabilirsiniz. Böylece C64'ün orijinal karakterlerini RAM'a aktarıp üstünde değişiklik yapabilirsiniz.

Şimdi gelelim önemli bir konuya....

Eğer LORAM, HIRAM ve CHAREN'in hepsini "0" yaparsanız ne olur ?.

Bunu yaparsanız artık \$D000 - \$DFFF arasındaki RAM'ı kullanabilir hale gelirsiniz. Ayrıca BASIC ve KERNAL ROM'ları devre dışı kalacağından C64'e adını veren 64 K RAM tamamen kullanıma açılmış olur. Fakat hiçbir giriş / çıkış işlemi yapılamayacağından ne işinize yarayacağını bilemem.

C64 normal bellek durumundayken BASIC ve KERNAL ROM adreslerine yapılacak bir POKE işlemiyle veri ROM'a değil RAM'e yazılacaktır. Zaten istersenizde ROM'a bir şey yazamazsınız. Çünkü ROM'un Türkçe'si YALNIZ OKUNABİLİR BELLEK'tir. Eğer PEEK komutuyla bir okuma işlemi yaparsanız elde edeceğiniz değer RAM'daki değil ROM'daki değerdir. Çünkü öncelik ROM'dadır.

1 numaralı adresi istediğiniz gibi kullanabilirsiniz ama dikkatli olmak şartıyla. Dergimizin ilk sayısında yer alan MAKİNE DİLİ EDİTÖRÜ çalışırken BASIC ROM altındaki RAM'de bulunur. Eğer MDE çalışırken bilgisayarı resetlerseniz aşağıdaki 5 komutu yazarak MDE'yi tekrar kullanabilirsiniz.

POKE 40956, 169
POKE 40957, 54
POKE 40958, 133
POKE 40959, 1
SYS 40956

Eğer 1 numaralı adresin içeriğinin \$37 olduğundan eminseniz,

POKE 40958, 198 POKE 40959, 1 SYS 40958 komutlarını yazmanız yeterlidir.

İlk dört POKE komutlarının assembler karşılığı şöyledir.

LDA #\$36 STA \$01

Son iki POKE komutunun assembler karşılığı ise DEC \$01 komutudur.

Bu iki adresten sonra sıra I / O bölgesi dediğimiz adreslere geldi. Genel bir ifadeyle \$D000 - \$DFFF adresleri arasında bulunan bu bölgede C64'ün görüntü işlemlerinden sorumlu VIC, ses işlemlerinden sorumlu SID, ekranda gördüğünüz renkler için kullanılan COLOR RAM, her türlü klavye, joystick, zamanlayıcı, kaset ve disket işlemlerinden sorumlu CIA1 ve CIA2 ile tamamen kullanıcıya ayrılmış durumda olan ve genellikle hardware için kullanılan IO1 ve IO2 bölgeleri vardır. Bazı kartuşlar bu adresleri kullanır. Eğer elektronik bilginiz varsa bu I/O1 ve I/O2 bölgelerini kullanarak bir takım devreler yapabilirsiniz.

Aşağıdaki tabloda her çipin yer aldığı adresleri göreceksiniz.

İSİM	GERÇEK İSMİ	ADRESİ	GERÇEK ADRESİ
VIC	Video Interface Controller	\$D000 - \$D3FF	\$D000 - \$D02E
SID	Sound Interface Device	\$D400 - \$D7FF	\$D400 - \$D41C
COLOR RAM	COLOR RAM	\$D800 - \$DBFF	\$D800 - \$DBE7
CIA1	Complex Interface Adapter 1	\$DC00 - \$DCFF	\$DC00 - \$DC0F
CIA2	Complex Interface Adapter 2	\$DD00 - \$DDFF	\$DD00 - \$DD0F
I / O 1	Input / Output 1	\$DE00 - \$DEFF	\$DE00 - \$DEFF
I / O 2	Input / Output 2	\$DF00 - \$DFFF	\$DF00 - \$DFFF

Tabloda iki ayrı adres görüyorsunuz I/O çiplerinin gerçek çalışma adresleri en sondaki sütundaki değerlerdir. C64'ü tasarlayanlar öyle bir iş yapmışki mesela ekranın dış rengi için kullanılan \$D020 adresi yerine \$D220 yazarsanız ekran rengi yine değişecektir. Bu durumda siz SHADOW REGISTER kullanmış oluyorsunuz. Yani yazdığınız adreste gerçekte hiç bir şey yoktur. Programlarınızda ilginçlik olsun diye kullanabilirsiniz. VIC çipinin registerleri her 64 adreste bir kendini tekrarlar. Yani \$D020 yerine \$D020 + 64 + 64 + kullanabilirsiniz. \$D020 için kullanabileceğiniz en son adres \$D3E0 olacaktır. Aynı tekrarlama SID çipi için 32 adrestir. CIA1 ve CIA2 çiplerinin registerleri ise her 16 adreste bir tekrarlanır. Yukarıdaki gereksiz bilgilerden sonra artık gerçek adresleri incelemeye başlayabiliriz.

ADRES		BİT	AÇIKLAMA
HEX	DEC		
\$D000 - \$D02E	53248 - 54271		MOS 6569 VIC (VIDEO INTERFACE CONTROLLER)
\$D000	53248		0 Numaralı SPRITE için X koordinatı değeri.
\$D001	53249		0 Numaralı SPRITE için Y koordinatı değeri.
\$D002	53250		1 Numaralı SPRITE için X koordinatı değeri.
\$D003	53251		1 Numaralı SPRITE için Y koordinatı değeri.
\$D004	53252		2 Numaralı SPRITE için X koordinatı değeri.
\$D005	53253		2 Numaralı SPRITE için Y koordinatı değeri.
\$D006	53254		3 Numaralı SPRITE için X koordinatı değeri.
\$D007	53255		3 Numaralı SPRITE için Y koordinatı değeri.
\$D008	53256		4 Numaralı SPRITE için X koordinatı değeri.
\$D009	53257		4 Numaralı SPRITE için Y koordinatı değeri.
\$D00A	53258		5 Numaralı SPRITE için X koordinatı değeri.
\$D00B	53259		5 Numaralı SPRITE için Y koordinatı değeri.
\$D00C	53260		6 Numaralı SPRITE için X koordinatı değeri.
\$D00D	53261		6 Numaralı SPRITE için Y koordinatı değeri.
\$D00E	53262		7 Numaralı SPRITE için X koordinatı değeri.
\$D00F	53263		7 Numaralı SPRITE için Y koordinatı değeri.
\$D010	53264		SPRITE'lar için 256'dan büyük X koordinatlarının ESB biti. Her bit bir sprite'a ayrılmıştır. Mesela 0 numaralı bit 0 numaralı SPRITE içindir.
\$D011	53265		VIC control registeri.
		7	Raster karşılaştırması 8. Bit. (\$D012'ye bakınız)
		6	Extented Color Mode : 1=Enable 0=Disable
		5	Bit Map Mode : 1=Enable 0=Disable
		4	Ekran açık / kapalı : 1=Açık 0=Kapalı
		3	24 / 25 satır seçimi : 0=24 Satır 1=25 Satır
		2-0	Y pixel konumuna (0-7) düz kaydırma
\$D012	53266		Karşılaştırma IRQ'su için raster değeri okuma / yazma
\$D013	53267		LIGHT PEN latch X konumu
\$D014	53268		LIGHT PEN latch Y konumu
\$D015	53269		Sprite Gösterme biti. Her bit bir sprite'a ayrılmıştır. Hangi bit "1" ise o bit'e ait sprite belirlenmiş koordinatlarda ekranda gözükür.
\$D016	53270		VIC kontrol registeri.
		7-6	Kullanılmıyor.
		5	BU BİTİ DAİMA "0" YAPIN!!
		4	Multi Color Mode : 1=Enable 0=Disable
		3	38 / 40 kolon seçimi : 0=38 Kolon 1=40 Kolon
		2-0	X konumuna düz kaydırma
\$D017	53271		Sprite'ları Y yönünde büyütme. Her bit bir Sprite için kullanılır. Hangi bit "1" ise o bit'e ait Sprite Y yönünde 2 kat büyük gösterilir.
\$D018	53272		VIC bellek kontrol registeri
		7-4	Ekran belleği taban adresi (VIC'in içinde)
		3-0	Karakter seti taban adresi (VIC'in içinde)
\$D019	53273		VIC interrupt bayrağı registeri.
		7	Herhangi bir açık IRQ koşulunda "1" olacak.
		6-4	Kullanılmıyor.
		3	LIGHT PEN interrupt bayrağı
		2	Sprite-Sprite çarpışması interrupt bayrağı
		1	Sprite-Karakter çarpışması interrupt bayrağı.
		0	Raster karşılaştırma interrupt bayrağı.
\$D01A	53274		Interrupt önleme registeri. Hangi interrupt kullanılacaksa ilgili bit "1" yapılmalıdır. Mesela Raster interruptı için 0 numaralı bit "1" olmalıdır.
\$D01B	53275		Sprite – karakter gösterme önceliği. Her bit bir Sprite'a ayrılmıştır. Hangi bit "1" ise o bite ait Sprite karakterlerin önünde gözükür.
\$D01C	53276		Sprite Multi Color Mode seçimi. Hangi bit "1" ise o bite ait Sprite Multi Color olarak gösterilir.
\$D01D	53277		Sprite'ları X yönünde büyütme. Her bit bir Sprite için kullanılır. Hangi bit "1"

			ise o bit'e ait Sprite X yönünde 2 kat büyük gösterilir.
\$D01E	53278		Yaratıklar arası çarpışmanın algılanması.
\$D01F	53279		Yaratık – zemin çarpışmasının algılanması.
\$D020	53280		Çerçeve rengi.
\$D021	53281		Zemin rengi 0.
\$D022	53282		Zemin rengi 1.
\$D023	53283		Zemin rengi 2.
\$D024	53284		Zemin rengi 3.
\$D025	53285		Multi Color Sprite'lar için renk registeri 0.
\$D026	53286		Multi Color Sprite'lar için renk registeri 1.
\$D027	53287		0. Sprite için renk registeri.
\$D028	53288		1. Sprite için renk registeri.
\$D029	53289		2. Sprite için renk registeri.
\$D02A	53290		3. Sprite için renk registeri.
\$D02B	53291		4. Sprite için renk registeri.
\$D02C	53292		5. Sprite için renk registeri.
\$D02D	53293		6. Sprite için renk registeri.
\$D02E	53294		7. Sprite için renk registeri.

VIC denilen çip aklınızdan ucundan bile geçmeyecek demo tekniklerinin baş sorumlusudur.
\$D011 - \$D012 - \$D016 - \$D018 registerleri bir araya geldimi VIC'i kimse tutamaz. ☺☺☺

Sırada C64'ün mükemmel seslerinden sorumlu olan SID çipi var.

ADRES		BİT	AÇIKLAMA
HEX	DEC		
\$D400 - \$D41F	54272 – 54300		MOS 6581 SID (SOUND INTERFACE DEVICE)
\$D400	54272		SES 1 : Frekans kontrolü küçük basamak baytı.
\$D401	54273		SES 1 : Frekans kontrolü büyük basamak baytı.
\$D402	54274		SES 1 : Darbe Dalga Genişliği küçük basamak baytı.
\$D403	54275	7-4	Kullanılmıyor.
		3-0	SES 1 : Darbe Dalga Genişliği üst yarım bayt.
\$D404	54276		SES 1 : Kontrol registeri.
		7	Rastgele Gürültü Dalga Bıçımı, 1 = ON
		6	Darbe Dalga Bıçımı, 1 = ON
		5	Testere Dişi Dalga Bıçımı, 1 = ON
		4	Üçgen Dalga Bıçımı, 1 = ON
		3	Test Bit'i : 1 = Osilatör 1'in çalışmasını durdurur.
		2	Osilatör 1 ile Osilatör 3'ün modüle edilmesi, 1 = ON
		1	Osilatör 1 ile Osilatör 3 frekanslarının eşzamanlı hale getirilmesi 1 = ON
		0	Geçit Biti : 1 = Yükselme / Düşme / Durma durumu dizisine başlama, 0 = Kaybolma durumuna geçiş.
\$D405	54277		Envelope Üretici 1 : Yükselme / Düşme Döngü Kontrolü
		7-4	Yükselme döngüsü süresi seçimi : 0 – 15
		3-0	Düşme döngüsü süresi seçimi : 0 – 15
\$D406	54278		Envelope Üretici 1 : Durma / Kaybolma Döngü Kontrolü
		7-4	Durma döngüsü süresi seçimi : 0 – 15
		3-0	Kaybolma döngüsü süresi seçimi : 0 – 15
\$D407	54279		SES 2 : Frekans kontrolü küçük basamak baytı.
\$D408	54280		SES 2 : Frekans kontrolü büyük basamak baytı.
\$D409	54281		SES 2 : Darbe Dalga Genişliği küçük basamak baytı.
\$D40A	54282	7-4	Kullanılmıyor.
		3-0	SES 2 : Darbe Dalga Genişliği üst yarım bayt.
\$D40B	54283		SES 2 : Kontrol registeri.
		7	Rastgele Gürültü Dalga Bıçımı, 1 = ON
		6	Darbe Dalga Bıçımı, 1 = ON
		5	Testere Dişi Dalga Bıçımı, 1 = ON

		4	Üçgen Dalga Biçimi, 1 = ON
		3	Test Bit'i : 1 = Osilatör 2'nin çalışmasını durdurur.
		2	Osilatör 2 ile Osilatör 1'in modüle edilmesi, 1 = ON
		1	Osilatör 2 ile Osilatör 1 frekanslarının eşzamanlı hale getirilmesi 1 = ON
		0	Geçit Biti : 1 = Yükselme / Düşme / Durma durumu dizisine başlama, 0 = Kaybolma durumuna geçiş.
\$D40C	54284		Envelope Üretici 2 : Yükselme / Düşme Döngü Kontrolü
		7-4	Yükselme döngüsü süresi seçimi : 0 – 15
		3-0	Düşme döngüsü süresi seçimi : 0 – 15
\$D40D	54285		Envelope Üretici 2 : Durma / Kaybolma Döngü Kontrolü
		7-4	Durma döngüsü süresi seçimi : 0 – 15
		3-0	Kaybolma döngüsü süresi seçimi : 0 – 15
\$D40E	54286		SES 3 : Frekans kontrolü küçük basamak baytı.
\$D40F	54287		SES 3 : Frekans kontrolü büyük basamak baytı.
\$D410	54288		SES 3 : Darbe Dalga Genişliği küçük basamak baytı.
\$D411	54289	7-4	Kullanılmıyor.
		3-0	SES 3 : Darbe Dalga Genişliği üst yarım baytı.
\$D412	54290		SES 3 : Kontrol registeri.
		7	Rastgele Gürültü Dalga Biçimi, 1 = ON
		6	Darbe Dalga Biçimi, 1 = ON
		5	Testere Dişi Dalga Biçimi, 1 = ON
		4	Üçgen Dalga Biçimi, 1 = ON
		3	Test Bit'i : 1 = Osilatör 3'in çalışmasını durdurur.
		2	Osilatör 3 ile Osilatör 2'nin modüle edilmesi, 1 = ON
		1	Osilatör 3 ile Osilatör 2 frekanslarının eşzamanlı hale getirilmesi 1 = ON
		0	Geçit Biti : 1 = Yükselme / Düşme / Durma durumu dizisine başlama, 0 = Kaybolma durumuna geçiş.
\$D413	54291		Envelope Üretici 3 : Yükselme / Düşme Döngü Kontrolü
		7-4	Yükselme döngüsü süresi seçimi : 0 – 15
		3-0	Düşme döngüsü süresi seçimi : 0 – 15
\$D414	54292		Envelope Üretici 3 : Durma / Kaybolma Döngü Kontrolü
		7-4	Durma döngüsü süresi seçimi : 0 – 15
		3-0	Kaybolma döngüsü süresi seçimi : 0 – 15
\$D415	54293		Filtrenin Geçiş Frekansı : Alt Yarım Bayt (2-0 bit'ler)
\$D416	54294		Filtrenin Geçiş Frekansı : Büyük Basamak Bayt'ı
\$D417	54295		Filtrenin salınım (Rezonans) kontrolü / Ses giriş kontrolü
		7-4	Filtre salınımı seçimi : 0 – 15
		3	Filtreye dışarıdan giriş : 1 = Evet 0 = Hayır
		2	Ses 3 filtre çıkışı : 1 = Evet 0 = Hayır
		1	Ses 2 filtre çıkışı : 1 = Evet 0 = Hayır
		0	Ses 1 filtre çıkışı : 1 = Evet 0 = Hayır
\$D418	54296		Filtre Mod ve Volüm Seçimi
		7	Ses 3 çıkışı kesimi : 1 = OFF 0 = ON
		6	Yüksek frekans geçişli filtre modu seçimi, 1 = ON
		5	Orta frekans geçişli filtre modu seçimi, 1 = ON
		4	Alçak frekans geçişli filtre modu seçimi, 1 = ON
		3-0	Çıkış ses volüm seçimi : 0 – 15
\$D419	54297		Analog Dijital çevirici : Game Paddle 1 (0 – 255)
\$D41A	54298		Analog Dijital çevirici : Game Paddle 2 (0 – 255)
\$D41B	54299		Osilatör 3 rastgele sayı üretici
\$D41C	54300		Ses üretici 3 çıkışı
\$DC00 - \$DC0F	56320 – 56335		MOS 6526
\$DC00	56320		CIA #1 (COMPLEX INTERFACE ADAPTER #1)
			Veri portu A (Klavye, Joystick, Paddles, Light-Pen)
		7-0	Klavye taraması için, klavye kolan değerleri yazımı
		7-6	Port A/B 'de bulunan paddle'ların okunması. (01=Port A, 10=Port B)
		4	Joystick A ateş düğmesi : 1= Ateşe basıldı
		3-2	Paddle ateş düğmeleri

		3-0	Joystick A yönü (0-15)
\$DC01	56321		Veri portu B (Klavye, Joystick, Paddles) Oyun Port'u 1
		7-0	Klavye taraması için, klavye sıra değerleri okuması
		7	B zamanlayıcısı : Toggle / Vuruş çıkışı
		6	A zamanlayıcısı : Toggle / Vuruş çıkışı
		4	Joystick 1 ateş düğmesi : 1= Ateşe basıldı
		3-2	Paddle ateş düğmeleri
		3-0	Joystick 1 yönü (0-15)
\$DC02	56322		Veri yön registeri – Port A (56320)
\$DC03	56323		Veri yön registeri – Port B (56321)
\$DC04	56324		Zamanlayıcı A : Alt Bayt
\$DC05	56325		Zamanlayıcı A : Yüksek Bayt
\$DC06	56326		Zamanlayıcı B : Alt Bayt
\$DC07	56327		Zamanlayıcı B : Yüksek Bayt
\$DC08	56328		Gün zamanı saati : 1 / 10 saniye
\$DC09	56329		Gün zamanı saati : Saniye
\$DC0A	56330		Gün zamanı saati : Dakika
\$DC0B	56331		Gün zamanı saati : Saat + Öğleden önce / Öğleden sonra (AM/PM) bayrağı (Bit 7)
\$DC0C	56332		Eş zamanlı veri G / Ç veri tamponu
\$DC0D	56333		CIA kesinti kontrol registeri (IRQ okuması / önleme yazımı)
		7	IRQ Bayrağı (1=IRQ oldu) / IRQ yap/yapma bayrağı
		4	Bayrak 1 IRQ (Kaset okuması / Seri bağlantı SRQ girişi)
		3	Seri bağlantı interrupt'ı
		2	Gün zamanı saati interrupt'ı
		1	B zamanlayıcısı interrupt'ı
		0	A zamanlayıcısı interrupt'ı
\$DC0E	56334		CIA kontrol register A
		7	Gün zamanı saati frekansı : 1 = 50 Hz 0 = 60 Hz
		6	Seri port G / Ç modu : 1 = Çıkış 0 = Giriş
		5	A zamanlayıcısı sayımı : 1 = CNT sinyalleri 0 = Sistem φ2 saati
		4	A zamanlayıcısı da yüklenecek mi ? 1 = Evet
		3	A zamanlayıcısı çalışma modu : 1 = 1 vuruşluk 0 = Sürekli
		2	A zamanlayıcısının PB6'ya çıkış modu 1 = Toggle 0 = Vuruş
		1	A zamanlayıcısının PB6'daki çıkışı 1 = Evet 0 = Hayır
		0	A zamanlayıcısının başlaması / durması : 1 = Başlama 0 = Durma
\$DC0F	56335		CIA kontrol register B
		7	Alarm / TOD saati kurmak 1 = Alarm 0 = Saat
		6-5	B zamanlayıcısının mod seçimi 00 : Sistem φ2 saat vuruşu sayımı 01 : Pozitif CNT geçişi sayımı 10 : A zamanlayıcısı "Underflow" vuruşu sayımı 11 : CNT pozitif iken A zamanlayıcısı underflow sayımı
		4-0	B zamanlayıcısı için CIA kontrol register'i A'nın aynısı
\$DD00- \$DD0F	56576- 56591		MOS 6526 CIA #2 (COMPLEX INTERFACE ADAPTER #2)
\$DD00	56576		Veri Port'u A (Seri bağlantı RS-232, VIC Bellek kontrolü)
		7	Seri bağlantı veri girişi
		6	Seri bağlantı saat vuruşu girişi
		5	Seri bağlantı veri çıkışı
		4	Seri bağlantı saat vuruşu çıkışı
		3	Seri bağlantı ATN çıkış sinyali
		2	RS-232 Veri çıkışı (Kullanıcı Port'u)
		1-0	VIC Çip'i sistem bellek bankası seçimi (Default = 11)
\$DD01	56577		Veri Port'u B (Kullanıcı Port'u, RS-232)
		7	Kullanıcı / RS-232 Teyp hazır
		6	Kullanıcı / RS-232 Gönderime hazır
		5	Kullanıcı

		4	Kullanıcı / RS-232 Taşıyıcı tarama
		3	Kullanıcı / RS-232 Halka göstergeç
		2	Kullanıcı / RS-232 Veri terminali hazır
		1	Kullanıcı / RS-232 Gönderim isteği
		0	Kullanıcı / RS-232 Alınmış veri
\$DD02	56578		Veri yön register'i : Port A
\$DD03	56579		Veri yön register'i : Port B
\$DD04	56580		A zamanlayıcısı alt bayt
\$DD05	56581		A zamanlayıcısı üst bayt
\$DD06	56582		B zamanlayıcısı alt bayt
\$DD07	56583		B zamanlayıcısı üst bayt
\$DD08	56584		Gün saati : 1/10 saniye
\$DD09	56585		Gün saati : saniye
\$DD0A	56586		Gün saati : dakika
\$DD0B	56587		Gün saati : saat + AM/PM bayrağı (bit 7)
\$DD0C	56588		Eş zamanlı Seri G/Ç veri tamponu
\$DD0D	56589		CIA interrupt kontrol register'i (NMI okuma/yazma maskesi)
		7	NMI bayrağı (1=NMI oluşumu) / 1-0 bayrağı ???
		4	FLAG1 NMI (Kullanıcı / RS-232) alınmış veri girişi
		3	Seri Port Interrupt'ı
		1	B zamanlayıcısı interrupt'ı
		0	A zamanlayıcısı interrupt'ı
\$DD0E	56590		CIA Kontrol Register A
		7	Gün saati frekansı : 1 = 50 Hz 0 = 60 Hz
		6	Seri Port G/Ç modu : 1 = Çıkış 0 = Giriş
		5	A zamanlayıcısı sayacı : 1 = CNT sinyali 0 = Sistem ϕ 2 saati
		4	A zamanlayıcısına zorunlu yükleme : 1 = Evet
		3	A zamanlayıcısına RUN modu : 1 = Tek atış 0 = Devamlı
		2	PB6'da A zamanlayıcısı Çıkış Modu : 1 = Toggle 0 = Vuruş
		1	PB6'da A zamanlayıcısı Çıkışı : 0 = Evet 1 = Hayır
		0	A zamanlayıcısı Başlatma / Durdurma : 1 = Başlatma 0 = Durdurma
\$DD0F	56591		CIA kontrol register B
		7	Alarm / TOD saatini tanımla 1 = Alarm 0 = Saat
		6-5	B zamanlayıcısının mod seçimi 00 : Sistem ϕ 2 saat vuruşlarını say 01 : Pozitif CNT geçişlerini say 10 : A zamanlayıcısı "Underflow" vuruşlarını say 11 : CNT pozitif iken A zamanlayıcısı underflow'larını say
\$DE00- \$DEFF	56832- 57087		I/O 1 Gelecekteki G/Ç genişlemeleri için ayrılmıştır.
\$DF00- \$DFFF	57088- 57343		I/O 2 Gelecekteki G/Ç genişlemeleri için ayrılmıştır.

I/O1 ve I/O2 denilen adres bölgelerini kullanarak C64'e yeni donanımlar eklenebilir. Bu donanımlar C64'e EXPANSION PORT'tan takılır. Bazı hızlandırıcı kartuşlar mesela -MULTI ICE 6 kendi üstündeki 32 K EPROM ve 8 K RAM'i seçmek için IO1 ve IO2 bölgesini kullanır. C64'e Harddisk ve CD-ROM bağlamanızı sağlayan [IDE64](#) isimli kartuş, REU (Ram Expansion Unit), Z80 kartuşu, çeşitli MIDI interface kartuşları hep bu bölgeleri kullanır. Eğer elektronik (dijital) bilginiz varsa bu bölgeleri kullanarak Analog / Dijital, Dijital / Analog Çeviriciler, Röle Kontrol Devreleri, Eprom Programlayıcılar vs. yapabilirsiniz.

Böyle bir işe kalkışmadan önce yanlış bir bağlantı sonucu C64'ünüzün çöpe gidebileceğini unutmayın. C64'e yeni donanım eklemek için USER PORT'u hatta JOYSTICK PORT'larını bile kullanabilirsiniz. Eğer joystick portlarını kullanırsanız klavyeyi kullanamayabilirsiniz.. Çünkü joystick portları ile klavye aynı adresleri kullanır. Yapacağınız donanımı programla kontrol edeceğiniz için bu sorunu halledebilirsiniz. Hangi port'u kullanırsanız kullanın iyi bir elektronik bilgisine sahip olmanız gerekir. Eğer yaptığınız ve çalışır projeler varsa dergiye gönderebilirsiniz.