

Eğitim Amaçlı Temel Bir Mikrobilgisayarın Tasarlanması ve Gerçekleşmesi

Design and Implementation of a Basic Microcomputer for Educational Purpose

Mehmet BAKACAK, Taner TOPAL

Bilgisayar Mühendisliği
Kırıkkale Üniversitesi

m.bakacak@gmail.com, ttopal@kku.edu.tr

Özet

Günümüzde elektronik araç gereçlerin çoğu en az bir adet mikrobilgisayar içerir. Mikrobilgisayarlar elektronik cihazların çalışmalarını kontrol eder ve istenilen işlemleri yapmasını sağlar. Mikrobilgisayarlar ihtiyaçlara göre sürekli gelişen ve değişen bir yapıya sahiptir. Mikrobilgisayarların yapısı ve çalışması hakkında gerekli bilgi beceriyi kazandırmak teorik olarak zordur. Eğitim ortamında mikrobilgisayarları daha iyi anlatabilmek için somutlaştırmak gerekir. Bu çalışmanın amacı da temel bir mikrobilgisayarı basite indirgeyerek tasarlamak ve uygulamaktır. Böylece eğitimde anlaşılmasını kolaylaştırmaktır. Çalışmada konu edilen mikrobilgisayar, Morris Mano'nun Bilgisayar Sistemi Mimarisi (Üçüncü baskı) kitabının [1] beşinci bölümünde tanımlanmaktadır. Tek çevrim, 16 bit veri yolu, 12 bit adres yolu ve 8 Kilobyte bellek boyutuna sahiptir. Bu mikrobilgisayar VHDL dilinde tasarlanmış ve FPGA üzerinde uygulaması yapılmıştır.

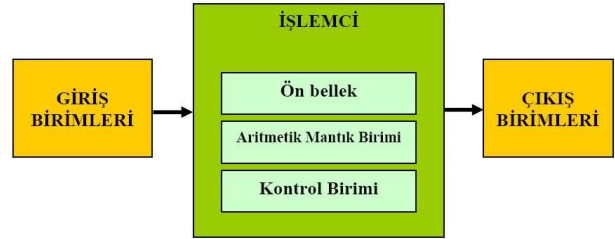
Abstract

Today, most of electronic equipments includes at least one microcomputer. Microcomputers control the operation of electronic devices and to provide the required operations. The microcomputer has a constantly developing and changing structure according to the needs. To gain necessary information and skill on the structure and operation of the microcomputer is theoretically difficult. In the educational environment, the microcomputer need to embody to give a better understanding. The purpose of this study is design and implement a basic microcomputer by simplifying. Thus, facilitate understanding in education. The microcomputer which was the subject of this study is defined in the fifth section of Morris Mano's computer system architecture book [1] (third edition). Is a Single cycle and has 16 bit data bus, a 12 bit address bus and a memory size of 8 kilobytes. This microcomputer is designed on VHDL and reviewed the implementation on FPGA.

1. Giriş

Bir mikrobilgisayar farklı görevleri icra eden bir dizi birimden meydana gelmektedir. Her birimin giriş ve çıkışlarına göre bir davranışı mevcuttur. Bu davranışları oluşturan yine bu birimlerin içinde bir dizi elektronik devredir.

İşlemci (Central Processing Unit (CPU) - Merkezi İşlem Birimi), Hafıza (Memory) ve Giriş / Çıkış Birimlerini (Input / Output Units) bulunduran yapının bütününe mikrobilgisayar denir. Şekil 1'de bir mikrobilgisayarın birimleri gösterilmektedir.

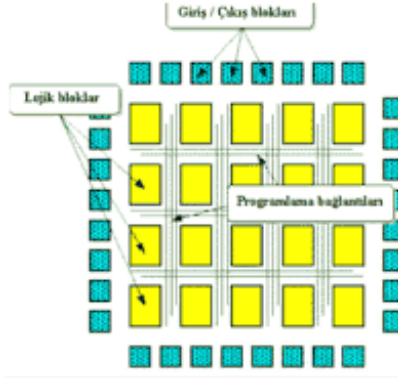


Şekil 1: Temel mikrobilgisayar yapısı.

Bir mikrobilgisayarı devre olarak tasarlayıp test etmek oldukça zaman alır ve maliyetlidir. Ayrıca tasarım aşamasında yaşanabilecek aksaklıklar, hatalar tasarımın en baştan kontrolünü gerektireceği için maliyet ve zaman açısından büyük kayıplara sebep olacaktır. Geleneksel olarak kullanılan yöntemler ile bu devreleri tasarlamak oldukça zor olduğu için artık yeni teknolojiler geliştirilmektedir. FPGA (Field Programmable Gate Array - Sahada Programlanabilir Kapı Dizileri) 'lar da bu teknolojilerden biridir.

FPGA, ABD'de 1984 yılında Xilinx'in kurucusu Ross Freeman tarafından icat edilmiştir [2]. Sonrasında Altera firması da FPGA alanında ürünler geliştirilmiştir. FPGA'ların mimari yapısı Şekil 2'de gösterildiği gibi üç bloktan oluşur. Bunlar;

- Lojik Bloklar (Configurable Logic Block CLB)
- Giriş - Çıkış Blokları (I / O)
- Bağlantı Blokları



Şekil 2: FPGA mimarisi.

FPGA'ye sahada programlanabilir ismi verilmesinin nedeni, mantık bloklarının ve ara bağlantıların üretimi sürecinden sonrasında da programlanabiliyor olmasıdır. FPGA'ler donanım tanımlama dilinden biri kullanılarak programlanabilirler. Kullanılan FPGA platformuna bağlı olarak, tasarlanan programları derlemek için Xilinx'de ISE, Altera'da ise Quartus programı kullanılır.

FPGA programlamak için tasarım yöntemi olarak Grafikselle (Şematik) tasarım ve HDL (Hardware Description Language - Donanım Tanımlama Dili) tasarım olmak üzere iki farklı yöntem kullanılır.

Grafiksel / Şematik tasarımda, tasarım derleyici programının (ISE, Quartus vs) kütüphanesinde yer alan araçlar ve mantık kapıları kullanılarak yapılır. HDL tasarımda ise tasarımın, HDL dillerinden her hangi bir tanesi kullanılarak yapılır. VHDL (Very High Speed Integrated Circuit Hardware Description Language - Yüksek Hızlı Tümlşik Devreler için Donanım Tanımlama Dili) ile Verilog en yaygın kullanılan iki dil türüdür. Bu çalışmada ise VHDL dili kullanılmıştır.

Çalışmada konu edilen mikrobilgisayar, Morris Mano'nun Bilgisayar Sistemi Mimarisi (Üçüncü baskı) kitabının [1] beşinci bölümünde tanımlanmaktadır. Tasarımı anlatılan mikrobilgisayar tek çevrim, 16 bit veri yolu, 12 bit adres yolu ve 8 Kilobyte bellek boyutuna sahiptir.

Mano'nun tasarımını temel alan benzer çalışmalar incelenmiş fakat tüm tasarımı eksiksiz içeren çalışmanın yapılmadığı gözlenmiştir. Tasarlanan bu mikrobilgisayar ile Mano'nun tanımladığı yapı ve diğer eklemelerle birlikte tamamlanmıştır.

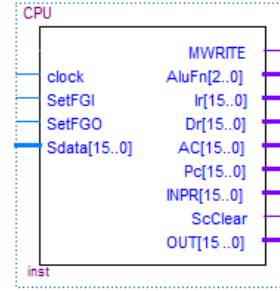
Tasarlanan bu mikrobilgisayar VHDL dilinde oluşturulmuş ve FPGA üzerinde uygulaması yapılmıştır. Uygulama yapılırken, Altera firmasının üretmiş olduğu DE0 Nano FPGA kartı [3] kullanılmıştır. Derleyici olarak Altera'nın Quartus II Web Edition [4], simülör olarak ise Modelsim Altera [5] kullanılmıştır.

2. Tasarlanan Mikrobilgisayarın Özellikleri

Bu bölümde, tasarlanan mikrobilgisayarın özelliklerinden bahsedilecektir. İlk olarak mikrobilgisayarın mimarisi anlatılacaktır. Ardından veri yolu yapısı, kaydedicileri, komut seti, aritmetik mantık birimi ve bellek yapısı hakkında bilgi verilecektir.

2.1. Mimarisi

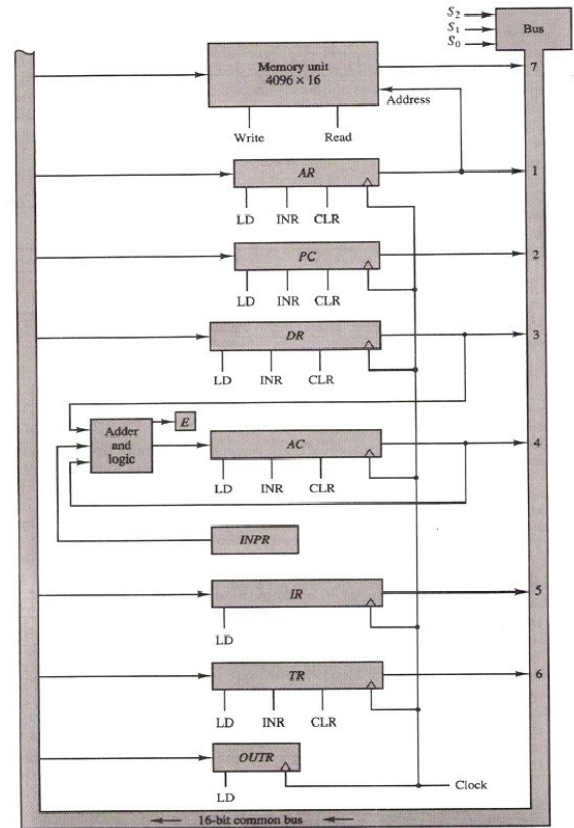
Tasarlanan mikrobilgisayarın en üst mimarisi Şekil 3'de de görüldüğü gibi dört adet giriş portu ve 9 adet çıkış portuna sahiptir. Mimari üç adet alt ünitenin birleşiminden meydana gelmektedir. Bunlar zamanlayıcı ünitesi, kontrol ünitesi ve kaydedicilerin, ALU ve belleğin yer aldığı tanımlama ünitesinden oluşmaktadır. Bu alt ünitelerde ayrıca kendi alt birimlerinden oluşmaktadır. Mikrobilgisayarda bütün komutlar tek saat çevriminde (single cycle) ve saat darbesinin yükselen kenarında işlem görmektedir.



Şekil 3: Mikrobilgisayarın en üst mimarisi.

2.2. Genel Veri Yolu Yapısı

Tasarlanan mikrobilgisayar sekiz adet kaydedici (register), bellek birimi ve kontrol biriminden oluşmaktadır. Birimler kendi arasında Şekil 4'de [1] de görüldüğü gibi 16 bitlik veri yoluyla birbirlerine bağlanmıştır. Veri yolu seçimi için 3 bitlik seçme sinyali tanımlanmıştır. Şekil 7'de veri yolu seçimi için tanımlama kodları gösterilmektedir.



Şekil 4: Genel veri yolu yapısı.

2.3. Kaydediciler

Kaydediciler (Registers) mikroişlemciler içerisinde yer alan küçük depolama birimleridir. Her birinin ayrı görevleri vardır. Kontrol biriminde ve aritmetik işlemlerde aktif görev alırlar. Tasarlanan bu mikrobilgisayarın içerisinde yer alan kaydediciler Çizelge 1’de gösterilmiştir.

Çizelge 1: Tasarlanan mikrobilgisayarın kaydedici listesi

Sembölü	Bit Sayısı	Adı
DR	16	Data register
AR	12	Address register
AC	16	Accumulator
IR	16	Instruction register
PC	12	Program counter
TR	16	Temporary register
INPR	16	Input register
OUTR	16	Output register

2.4. Komut Seti

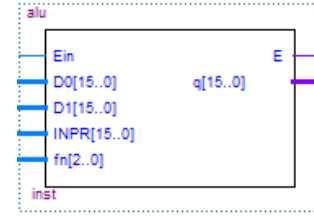
Bir mikroişlemcinin neler yapabildiğini komut seti (instruction set) belirler. Tasarlanan bu mikrobilgisayarın komut seti Çizelge 2’de [1] gösterilmektedir.

Çizelge 2: Tasarlanan mikrobilgisayarın komut seti

Symbol	Hexadecimal code		Description
	I = 0	I = 1	
AND	0xxx	8xxx	AND memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load memory word to AC
STA	3xxx	Bxxx	Store content of AC in memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA		7800	Clear AC
CLE		7400	Clear E
CMA		7200	Complement AC
CME		7100	Complement E
CIR		7080	Circulate right AC and E
CIL		7040	Circulate left AC and E
INC		7020	Increment AC
SPA		7010	Skip next instruction if AC positive
SNA		7008	Skip next instruction if AC negative
SZA		7004	Skip next instruction if AC zero
SZE		7002	Skip next instruction if E is 0
HLT		7001	Halt computer
INP	F800		Input character to AC
OUT	F400		Output character from AC
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off

2.5. Aritmetik Lojik İşlem Birimi (ALU)

Tasarlanan mikrobilgisayardaki ALU’nun şematik gösterimi Şekil 5’deki gibidir. 16 bitlik giriş kaydedici (INPR), yine 16 bitlik D0 (DR’den gelen) ve D1 (veri yolundan gelen), 1 bitlik elde biti (Ein) ve ALU komutlarının seçimi için 3 bitlik fn den oluşan giriş portlarına sahiptir. 1 bitlik işlemler sonucu oluşan elde biti (E) ve 16 bitlik veri çıkış (q) çıkış portuna sahiptir.



Şekil 5: Aritmetik mantık birimi.

Tasarlanan mikrobilgisayardaki ALU’nun VHDL dilindeki kodları Şekil 6’da gösterilmiştir. Şekilde de görüldüğü gibi bazı komutların gerçekleştirilmesi için alt çipler tanımlanmıştır. ALU’nun 3 bitlik fn portundaki seçim değerine göre işlem yapılır. Yapılacak işlem, ilgili alt çipe yönlendirilir. Elde edilen sonuçlar ALU’nun çıkış portuna gönderilir.

```

4 entity alu is
5 port(
6   Ein: in std_logic;
7   D0, D1: in Tword;
8   INPR: in Tword;
9   fn : in aluinstr;
10  E: out std_logic;
11  q: out Tword
12 );
13 end alu;
14
15 architecture a of alu is
16   signal Cout1,Cout2,Cout3: std_logic;
17   signal notD1, and2, add2, shr, shl, notINPR: Tword;
18 begin
19   chip1: andnbit generic map(n=>16) port map(D0, D1, and2);
20   chip2: fan generic map(n=>16) port map(D0, D1, add2,Cout1);
21   chip3: shift generic map(n=>16) port map(D1,Ein,'0',shr,Cout2);
22   chip4: shift generic map(n=>16) port map(D1,Ein,'1',shl,Cout3);
23   chip5: com generic map(n=>16) port map(D1, notD1);
24
25   q <= D0 when fn=aluPass
26       else notINPR when fn=aluINPR
27       else and2 when fn=aluAND
28       else add2 when fn=aluADD
29       else notD1 when fn=aluCOM
30       else shr when fn=aluSHR
31       else shl when fn=aluSHL
32       else zeroword;
33
34   E <= Cout1 when fn=aluADD
35       else Cout2 when fn=aluSHR
36       else Cout3 when fn=aluSHL
37       else '0';
38 end a;

```

Şekil 6: ALU’nun VHDL kodu.

Tasarlanan mikrobilgisayardaki ALU komutları ve veri yolu seçim kodlarının tanımlandığı paket Şekil 7’de gösterilmektedir.

```

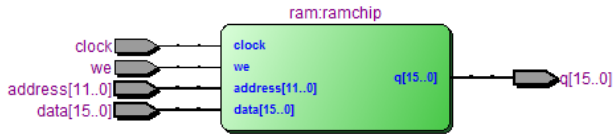
1 library ieee; USE ieee.std_logic_1164.all;
2
3 PACKAGE defines IS
4   -- Subtypes
5   subtype aluinstr is std_logic_vector(2 downto 0);
6   subtype busctrl is std_logic_vector(2 downto 0);
7
8   -- Constants
9   -- ALU instructions
10  constant aluPass: aluinstr := "000";
11  constant aluAND: aluinstr := "001";
12  constant aluADD: aluinstr := "010";
13  constant aluCOM: aluinstr := "011";
14  constant aluSHR: aluinstr := "100";
15  constant aluSHL: aluinstr := "101";
16  constant aluINPR: aluinstr := "110";
17
18  -- Bus select codes
19  constant BusAR: busctrl := "001";
20  constant BusPC: busctrl := "010";
21  constant BusDR: busctrl := "011";
22  constant BusAC: busctrl := "100";
23  constant BusIR: busctrl := "101";
24  constant BusTR: busctrl := "110";
25  constant BusMem: busctrl := "111";
26
27 end defines;

```

Şekil 7: ALU komutları ve veri yolu seçme kodları.

2.6. Bellek Yapısı

Tasarlanan mikrobilgisayarın bellek biriminin yapısı Şekil 8’de görülmektedir. 16 bitlik veri girişi ile veri yoluna bağlanmıştır. Adres genişliği 12 bitlik olup adres kaydediciye bağlanmıştır. Bellek boyutu $2^{12} = 4096$ kelime uzunluğuna sahiptir ve her bir kelime 16 bit içermektedir. Yazma (we) sinyali aktif olduğunda belleğe yazma işlemi gerçekleştirilir. Diğer durumlardan bellekten veri okunmaktadır. Belleğe CPU’nun yapması gereken program dışarıdan yüklenmektedir. Bellek gerekli bilgileri kendi üzerinden okumakta ve yazmaktadır. Bellek içerisinde 16 bitlik veri ikili kodla ilgili adrese yazılacak şekilde tanımlanmıştır.



Şekil 8: Bellek birimi.

3. Tasarlanan Mikrobilgisayarın Programlanması

Tasarlanan mikrobilgisayarı test etmek için örnek bir program yazılmıştır. Program Quartus programında derlenerek Modelsim - Altera üzerinde simülasyonu yapılmıştır. Sonuçlar kontrol edildikten sonra FPGA üzerine tasarlanan mikrobilgisayar yüklenmiş ve programın çıktıları gözlenmiştir.

3.1. Örnek Program

İki sayının toplamını bulan ve sonucunu çıkış portuna gönderen programın kodlaması yapılmıştır. Onluk değeri 83 olan A sayısı ile onluk değeri -23 olan B sayısının toplamı, sonucunun çıkış portunda onluk 60 değeri elde edilmesi istenmiştir.

Bellek içerisinde bu program ikili kodla ilgili adrese yazılmıştır. Örneğin programın ilk satırında 0. adrese "0010000000000100" verisi yazılmıştır. Bu veri LDA komutunu işaret etmektedir.

Bellek Adresi=> "İkili Kodu" --Sembolik Kodu

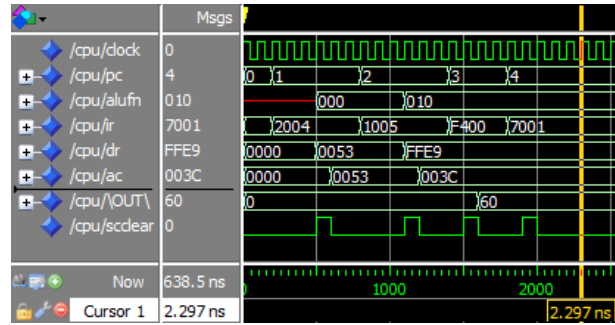
```
0 => "0010000000000100",-- LDA A
1 => "0001000000000101",-- ADD B
2 => "1111010000000000",-- OUT
3 => "0111000000000001",-- HLT
4 => "0000000001010011",-- A
5 => "111111111101001",-- B
```

Sembolik kod bellek içerisinde satırların neyi ifade ettiğini açıklama olması için eklenmiştir başka bir fonksiyonu yoktur.

3.2. Simülasyonu

Belleğe yazılan program Quartus programı [5] ile derlendikten sonra Modelsim - Altera uygulaması [6] kullanılarak sonuçlar gözlenmiştir. Şekil 9’da bölüm 3.1’de ki örnek programın simülasyon sonuçları görülmektedir. Anlaşılabilirliği kolaylaştırmak için burada PC ve OUT onluk biçiminde, IR, DR ve AC ise onaltılık sayı biçiminde ayarlanmıştır.

Simülasyonda da görüldüğü gibi çıkış portunda (OUT) beklenen değer olarak onluk 60 sayısı başarılı bir şekilde elde edilmiştir.



IOF kesme komutları da eklenmiştir. Ayrıca “Mano’s Micro”[7] çalışmasında Ram’e yazma yapılamamakta sadece okuma yapılabilmektedir. Tasarlanan bu mikrobilgisayarda okuma ve yazma işlemleri yapılabilecek şekilde tasarlanmıştır. “Mano’s Micro” [7] çalışmasında Input Register bulunmamaktadır. Tasarlanan bu mikrobilgisayara Input Register’da eklenmiştir.

Yapılan bu çalışma, Mano’nun tarif ettiği tüm komut setleri ve yapıları içerecek şekilde tasarlanmıştır. Bu tasarım üzerinde ayrıca iyileştirmelerde yapılmıştır. Örneğin RAM yapısı değiştirilerek aynı birimden okuma ve yazma işlemi başarılı bir şekilde çalıştırılmıştır. Input ve Output portu 16 bite çıkarılmıştır. Kullanılan FPGA kartının 8 bitlik çıkış portu bulunduğu için, 16 bitlik output portunu okuyabilmek için düşük ve yüksek 8 biti gösterilecek şekilde butonla seçimli hale getirilmiştir. “Mano’s Micro” [7] çalışmasında sadece işlem sonucu görülmektedir. Bu çalışmadaki mikrobilgisayarın da çok hızlı çalışmasından dolayı sonuçları gözlemleyebilmek için frekans bölücü kullanılmıştır ve saat frekansını 1 KHz – 50 MHz aralığında istenilen frekansta çalıştırılabilir hale getirilerek sonuçları istenilen hızda gözlenmesi sağlanmıştır.

Mikrobilgisayar tasarımı ve uygulaması gerçekte zaman ve maliyet açısından oldukça zor bir süreçtir. Bu yüzden de eğitimde uygulamaları çoğu zaman teoride kalmaktadır. Bu çalışma ile temel bir mikrobilgisayarın tasarlanması ve uygulanması yapılmıştır. Eğitim ortamı için somutlaştırılmış ve temel bir yapı kullanılarak anlaşılması kolaylaştırılmıştır. Programlama bilgi ve becerisine sahip öğrenciler için hem bir başlangıç hem de rehber olması sağlanmıştır. Bu sayede öğrenciler teoride anlatılan bir mikrobilgisayarı kendileri tanımlayabilecek ve çalışmasını gözlemleyebilecektir. Hatta kendilerini geliştirerek ihtiyaca uygun farklı bir mikrobilgisayarı en baştan tasarlayabileceklerdir.

5. Kaynaklar

- [1] M. Morris Mano, “Computer System Architecture (3rd edition)”, (2000).
- [2] <http://fpganedir.com/FPGA/xilinx.php>, (indirme tarihi: 05.07.2016)
- [3] Terasic Technologies, Terasic DE0-Nano User Manuel, (2012)
- [4] Altera Corporation, Quartus II Handbook Version 10.1 Volume 1: Design and Synthesis, (2010)
- [5] Altera Corporation, Getting Started with Quartus II Simulation Using the ModelSim-Altera Software, (2011)
- [6] Altera Corporation, Creating Testbench using ModelSim-Altera Wave Editor, (2011)
- [7] N. Narasimhamurthi, “Mano’s Micro”, Technical Not