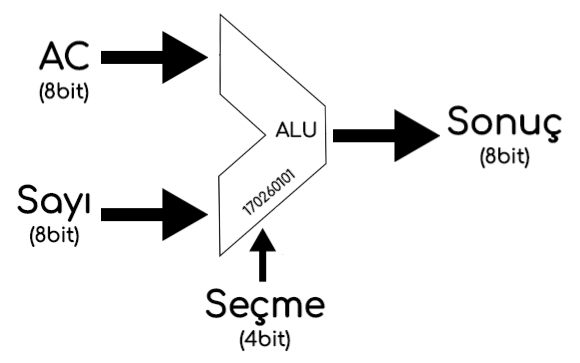
**1. Giriş**

Ödevde, programlanabilir FPGA ortamı için basit bir ALU tasarlanmıştır. FPGA programlama için, donanım programlama dili olarak Verilog tercih edilmiştir. Tasarlanan ALU için 16 farklı işlem belirlenmiş ve bu işlemler 4 bitlik işlem kodları ile tanımlanmıştır. Tasarlanan ALU toplama işlemi, çıkarma işlemi, çarpma işlemi, bölme işlemi, kaydırma işlemi, ve işlemi, veya işlemi, karşılaştırma işlemleri gibi temel bir çok işlevi yerine getirebilmektedir. Tasarım, ModelSim programı üzerinde emüle edilmiştir. Raporda işlem tanımları ve aşamaları tariflenmiş, şekil ve çizelgelerle açıklanmaya çalışılmıştır.

**2. Uygulama**

Ödevin gerçekleştirilmesinde internet kaynaklarından faydalanılarak Verilog syntax yapısı ve FPGA ortamında ALU tasarım süreci hakkında bilgiler edinilmiş ve uygulama safhasına geçilmiştir. Yararlanılan kaynaklar raporun sonunda ayrıca belirtilmiştir. HDL olarak Verilog dili, emüle ortamı için ise ModelSim programı kullanılmıştır.

Tasarlanılan basit ALU, 8 bitlik iki operandı işleme koyarak 8 bitlik sonuç değeri üretmektedir. Operandlar AC ve Sayi ve sonuç değeri Sonuç olarak tanımlanmıştır.



**-Şekil 1:** ALU Yapısı ve I/O Değişkenleri-

Bu ALU yapısı, 16 farklı işlem gerçekleştirebilmektedir. 16 farklı işlem 4 bitlik işlem kodları ile tanımlanmıştır. İşlem kodları ile, gerçekleştirilecek mantıksal ve aritmetiksel işlevlerin seçme işlemi gerçekleştirilmektedir.

Örneğin, tasarlanan yapıda 0000 işlem kodu toplama işlemini tariflemekte ve ilk operand ile ikinci operandı işleme alarak 8 bitlik sonuç değeri toplamlarını çıkışa vermektedir.

**AC =** 00000001, **Sayı =** hex 00000010, **Seçme =** 0000

Bu iki operand ve seçme değeri için işleme alınan iki değer ALU’da toplama gerçekleştirilerek çıkışta Sonuç değeri 00000011 üretilecektir.

Bu yapı için gerekli input ve output tanımlamaları şöyledir:

**input** [7:0] AC;

**input** [7:0] Sayi;

**input** [3:0] IsKodu;

**output reg** [7:0] Cikti;

Bu yapı için toplama işlemi tanımlaması şöyledir:

**if (**IsKodu == 4'b0000**)**

Cikti = AC + Sayi ;

Diğer işlem tanımlamaları için gerekli Verilog kodları rapordaki ekte yer almaktadır.

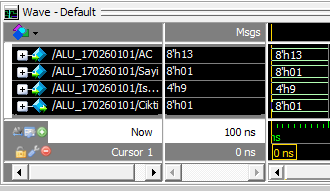
Aşağıdaki şekilde tasarlanan yapı için işlem tipleri, kodları ve gerçekleştirilen 16 adet farklı işlem yer almaktadır. Bu 16 farklı işlem 4 bitlik Seçme girişi ile belirlenecektir. İşlem kodları AC ve Sayi operandlarını işleme almaktadır.

|  |  |  |  |
| --- | --- | --- | --- |
| NO | İŞLEM TİPİ | İŞLEM KODU | İŞLEM |
| 1 | TOPLAMA | 0000 | AC + Sayi |
| 2 | ÇIKARMA | 0001 | AC - Sayi |
| 3 | ÇARPMA | 0010 | AC \* Sayi |
| 4 | BÖLME | 0011 | AC / Sayi |
| 5 | ARTTIRMA | 0100 | AC + 1 |
| 6 | TÜMLEME | 0101 | AC’ |
| 7 | 2’YE TÜMLEME | 0110 | AC’ + 1 |
| 8 | SOLA KAYDIRMA | 0111 | <<AC |
| 9 | SAĞA KAYDIRMA | 1000 | AC>> |
| 10 | VE (AND) | 1001 | AC ∧ Sayi |
| 11 | VEYA (OR) | 1010 | AC ∨ Sayi |
| 12 | VE DEĞİL (NAND) | 1011 | (AC ∧ Sayi)’ |
| 13 | VEYA DEĞİL (NOR) | 1100 | (AC ∨ Sayi)’ |
| 14 | ÖZEL VEYA (XOR) | 1101 | AC ^ Sayi |
| 15 | BÜYÜKLÜK KARŞILAŞTIRMA | 1110 | AC > Sayi |
| 16 | EŞİTLİK KARŞILAŞTIRMA | 1111 | AC == Sayi |

**-Şekil 2:** ALUİşlemleri ve İşlem Kodları-

**3. Sonuç**

Aşağıda, tasarlanan yapının ModelSim üzerinde emüle edilmiş hali yer almaktadır. Örnekte 00010011 (hex 13) ve 00000001 (hex 01) operandları için 1001 (hex 9) işlem kodu girilerek VE işlemi gerçekleştirilmiş ve 00000001 (hex 01) sonucu üretilmiştir.



**-Şekil 3:** ModelSim Programı Üzerinde Gösterim-

**4. Kaynaklar**

**[1]** http://www.asic-world.com/verilog/syntax2.html

**[2]** http://ee.sut.ac.ir/People/Courses/142/Summary%20of%20Verilog%20Syntax.pdf

**[3]** http://fpganedir.com/FPGA/fpga\_programlama.php

**[4]** http://www.fpganedir.com/simulasyon/modelsim\_proje.php

**[5]** https://www.fpga4student.com/2017/06/Verilog-code-for-ALU.html

**[6]** https://www.cizgi-tagem.org/wp-content/uploads/2014/02/Verilog-ile-FPGA-Tasarimina-Giris.pdf

**[7]** https://www.doulos.com/knowhow/verilog\_designers\_guide/if\_statement/