

PROBLEMİN SEÇİMİ

Bir ALU, bir yazmaç bloğu ve durum makinesinden ibaret bir kontrol birimi ile aşağıdaki listeden seçilecek bir problemin algoritmik çözümü istenmektedir.

Problems Listesi:

1. [1] nolu referansta açıklanan yöntemle, iki adet 4 bitlik pozitif tamsayının çarpımını hesaplama.
2. [1] nolu referansta açıklanan yöntemle, iki adet 4 bitlik 2'nin tümleyeni gösterimde işaretli tamsayının çarpımını hesaplama.
3. [2] nolu referansta açıklanan yöntemle, her ikisi de 2 bitlik pozitif tamsayı olan A ve B'yi kullanarak A^B 'yi hesaplama.
4. [3] nolu referansta açıklanan yöntemlerden biriyle, 8 bitlik pozitif A tamsayısı ve 4 bitlik pozitif B tamsayısını kullanarak A/B'yi hesaplama.
5. [4] nolu referansta açıklanan yöntemlerden biriyle, 8 bitlik pozitif A tamsayısını kullanarak \sqrt{A} 'yi hesaplama.
6. [5] nolu referansta açıklanan yöntemlerden biriyle, 8 bitlik pozitif A ve B tamsayılarını kullanarak $\text{mod}(A,B)$ 'yi hesaplama.
7. [6] nolu referanstan yararlanarak, 4 bitlik pozitif A ve B tamsayılarının geometrik ortalamasını hesaplama.
8. [7] nolu referanstan yararlanarak, (0,6) aralığında tamsayı olan A'nın faktoriyelini hesaplama.
9. [9] nolu referanstan yararlanarak, her ikisi de 8 bitlik pozitif tamsayı olan A ve B'yi kullanarak A-B'yi hesaplama.
10. Her ikisi de 8 bitlik pozitif tamsayı olan A ve B'yi kullanarak 2A ile B'yi karşılaştırma, küçükse 0, eşitse 1, büyükse 2 değerini çıkışta gösterme. [10] nolu referanstan faydalanabilirsiniz.
11. Her ikisi de 8 bitlik pozitif tamsayı olan A ve B'yi kullanarak A ile B/2'yi karşılaştırma, küçükse 0, eşitse 1, büyükse 2 değerini çıkışta gösterme. [10] nolu referanstan faydalanabilirsiniz.
12. [11] nolu referansta açıklanan yöntemle, iki 8 bitlik pozitif tamsayının en büyük ortak bölenini hesaplama.
13. Pozitif 8 bitlik iki tamsayının mutlak farkını hesaplama. [12] nolu referanstan faydalanabilirsiniz.
14. 2'nin tümleyeni gösteriminde işaretli 6 bitlik iki sayının mutlak farkını hesaplama. [12] nolu referanstan faydalanabilirsiniz.
15. Her ikisi de 8 bitlik pozitif tamsayı olan A ve B'yi kullanarak, $A=B$ ise $A+B$ 'yi, $A \neq B$ ise $A-B$ 'yi hesaplama. [12] nolu referanstan faydalanabilirsiniz.
16. Pozitif 8 bitlik iki tamsayıdan büyük olanının yarısını hesaplama. [13] nolu referanstan faydalanabilirsiniz.
17. [14] nolu referansta açıklanan yöntemle, her ikisi de 8 bitlik pozitif tamsayı olan A ve B'yi kullanarak, $2A+B/3$ 'ü hesaplama.
18. Pozitif 8 bitlik iki tamsayının aritmetik ortalamasının karesini hesaplama.
19. 2'nin tümleyeni gösteriminde işaretli 6 bitlik iki sayının aritmetik ortalamasının bulunması.
20. Her ikisi de pozitif 6 bitlik tamsayı olan A ve B'yi kullanarak $A-3B$ 'nin hesaplanması.

PROBLEMİN ÇÖZÜMÜ

Aşağıda tanımlanan kısıtlara uyarak seçtiğiniz problemi çözmeniz gerekmektedir.

Çözüm adımları:

1. Kısıtlarda verilen tanıma uygun ALU'yu tasarlayınız ve size verilen üst modüle yerleştiriniz.
2. Seçtiğiniz problemi yalnızca bu ALU'daki fonksiyonlarla çözecek algoritmayı tasarlayınız.
3. Algoritmanızı Algoritmik Durum Makinesi (Arithmetic State Machine, ASM) şeması ile Moore makine olarak çiziniz [15].
4. Davranışsal kodlama ile bu ASM'yi gerçekleştiren kontrol ünitenizi (CU) oluşturunuz. Çıkış tutucularınız tümüne istisnasız her durumda uygun atamayı yaptığınızdan emin olunuz.
5. Problem tanımınıza uygun test kodu hazırlayıp devrenizi test ediniz.
6. Test kodu ile üretilen dalga formunda devrenizin çalıştığını kanıtlayan ya da nasıl çalıştığını açıklayan kısımları yukarıdaki adım sonuçları ile birlikte raporunuza ekleyiniz. Bu dalga formu parçaları üzerine ok/yazı/sayı/şekil ekleyerek anlaşılmasını kolaylaştırabilirsiniz.
7. Kodlarınızın RTL şemalarını raporunuza ekleyiniz.

ÇÖZÜM KISITLARI

Çözümünüz aşağıdaki kısıtlara uyacak şekilde yapılmalıdır.

Kısıtlar:

1. Devrenizin TOP modülü Figür 1'de verilen şemadaki gibi olacaktır. Bu üç alt modülden başka bir devre parçası ya da modül olmayacaktır.
2. Figür 2'de verilen yazmaç bloğunun (Register Block, RB) kodu size örnek olarak verilmiştir. Bu örnekteki gibi ALU'nun bileşenlerini de ayrı ayrı modül olarak tasarlamanız gerekmektedir.
3. CU içinde modülün çıkış portlarını süren FF'lar ve durum kelimesini tutan FF'lardan başka FF olamaz.
4. Reset ve clk girişi CU ve RB için ortaktır. Reset asenkron aktif 1 olarak tasarlanacaktır.
5. CU içindeki çıkış portlarını süren FF'lar KEEP kısıtı ile korunacaktır. Bu sayede ISE'nin devrenizin kullanılmayan kısımlarını silmesi engellenecektir.
6. CU içerisindeki durum makinesi Moore devre olacaktır.
7. CU, "start"ın 1 olması ile çalışmaya başlayacak ve sonucu hesaplayıncaya kadar "busy" çıkışını 1 yapacaktır.

8. CU'da durum geçişlerini dallandıran koşul yalnızca ALU'dan gelen Carry-Out (CO) ve Zero (Z) işaretleri üzerinde tanımlanabilir, ör: if(CO), if(!Z).
9. Figür 3'e uygun ALU tasarlarken, aynen figürde görüldüğü gibi alt modülle kullanılacaktır. Kombinezonsal atama yaparsanız dahi bunu AND, XOR, ADD ya da Shift blokları içinde kullanınız.
10. Kodlardan üretilen RTL'ler verilen figürler ile tam uyum içinde olacaktır. Portları figürlerde verilen isimlerle isimlendiriniz.
11. ALU'nuz InsSel '00' iken AND, '01' iken XOR, '10' iken toplama yapacaktır. Bu figürden de anlaşılabilir. Toplama yaptığında taşma biti CO'ya atanacaktır. '11' olduğunda A girişindeki değer bir bit sola kaydırılacak, MSB, LSB'ye yazılacaktır ve CO'ya eski MSB atanacaktır. CO ilk iki durumda 0 olacaktır. Z işareti ise ancak ALU çıkışı 0 olduğunda 1 olacak, diğer durumlarda 0 olacaktır.
12. Kısıtlarda eksiklik, çelişki ya da hata farkedilmesi halinde Ninova üzerinden dersi veren araştırma görevlilerince duyuru yapılacaktır.

REFERANSLAR

- [1]<http://www4.wittenberg.edu/academics/mathcomp/shelburne/comp255/notes/BinaryMultiplication.pdf>
- [2]http://en.wikipedia.org/wiki/Exponentiation_by_squaring
- [3]http://en.wikipedia.org/wiki/Division_algorithm
- [4]http://en.wikipedia.org/wiki/Square_root_algorithm
- [5]<http://stackoverflow.com/questions/2773628/better-ways-to-implement-a-modulo-operation-algorithm-question>
- [6]http://en.wikipedia.org/wiki/Arithmetic-geometric_mean
- [7]<http://en.wikipedia.org/wiki/Factorial>
- [9]<http://www.cs.cornell.edu/~tomf/notes/cps104/twoscomp.html>
- [10]http://en.wikipedia.org/wiki/Digital_comparator
- [11]http://en.wikipedia.org/wiki/Euclidean_algorithm
- [12]<http://stackoverflow.com/questions/3878062/adding-and-subtracting-twos-complement>
- [13]<http://stackoverflow.com/questions/9954660/compare-two-binary-numbers-and-get-the-different-bits>
- [14]<http://courses.cs.vt.edu/~cs1104/BuildingBlocks/divide.030.html>
- [15]Brown&Vransic, Fundamentals of Digital Logic with Verilog Design, McGrawHill, 2002, Section 8.10.