

# İÇİNDEKİLER

İÇİNDEKİLER .....	ii
SİMGELER LİSTESİ .....	v
ŞEKİLLER LİSTESİ .....	vi
TABLolar LİSTESİ .....	viii
ÖNSÖZ .....	ix
ÖZET .....	x
ABSTRACT .....	xi
BÖLÜM 1	
GİRİŞ .....	1
BÖLÜM 2	
HÜCRESEL SİNİR AĞLARI .....	4
2.1. Temel Gösterim ve Tanımlamalar .....	5
2.1.1. Standart CNN Mimarisi .....	5
2.1.2. $(C(i, j))$ Hücresinin Etki Küresi .....	5
2.1.3. Normal ve Sınır Hücreleri .....	6
2.1.4. Standart HSA .....	6
2.1.5. Uzayda Değişmeyen yada Eşyönlü HSA .....	8
2.2 Matematiksel Temeller .....	8
2.2.1 Vektör, Matris Gösterimleri ve Sınır Koşulları .....	8
2.2.1.1. Sanal hücreler .....	9
2.2.1.2. Sınır Koşulları .....	9
2.2.1.3. Vektör fark denklemi .....	11
2.2.2 Uzayda Değişmeyen HSA .....	13
2.2.2.1. Klonlayan şablon gösterimi .....	13
2.2.3 Üç Basit HSA Sınıfı .....	16
2.2.3.1 Sıfır geri besleme (İleri Besleme) sınıfı $C(0, B, z)$ .....	16
2.2.3.2 Sıfır giriş (Autonomous) sınıfı $C(A, 0, z)$ .....	17
2.2.3.3 Ayrılmamış (Sayıl-Skaler) sınıf $C(A^0, B, z)$ .....	17
BÖLÜM 3	
GABOR SÜZGEÇLERİ VE HSA GERÇEKLEMELERİ .....	20
3.1 Giriş .....	20
3.2 Gabor Süzgeçleri .....	20
3.2.1 İki Boyutlu (2-D) Gabor Süzgeçleri .....	22
3.2.2 İki Boyutlu (2-D) Gabor Süzgecinin İşlemsel Ayrıntıları .....	24
3.3 Neden HSA Gabor Tip Süzgeç? .....	25
3.3.1 Bir Boyutlu HSA Gabor Tip Süzgeç .....	25
3.3.2 İki Boyutlu (2-D) HSA Gabor Tip Süzgeçler .....	31
3.3.3 İki Boyutlu (2-D) HSA Gabor Tip Süzgecin Dürtü Yanıtı, Frekans Yanıtı ve Bant Genişliği .....	36

## BÖLÜM 4

2 BOYUTLU HSA GABOR TİP SÜZGECİN DEVRE GERÇEKLEMELERİ .....	44
4.1 2 Boyutlu HSA Gabor Tip Süzgecin Analog Devre Gerçeklemesi .....	44
4.2 2 Boyutlu HSA Gabor Süzgecin Sayısal Gerçeklenmesi .....	47
4.2.1 2 Katmanlı 2 Boyutlu HSA GTS için Geri Euler Yaklaşıklığı .....	48

## BÖLÜM 5

FPGA'LERE GİRİŞ .....	52
5.1 FPGA'lere Genel Bakış .....	53
5.2 FPGA Programlama .....	54
5.2.1 Tasarım (Design entry) .....	55
5.2.1.1 Şematik tasarım (Schematic design entry) .....	55
5.2.1.2 Donanım Tanımlama Dilleri (Hardware Description Languages) (HDLs) .....	55
5.2.2 Netlist Gösterimi .....	58
5.2.3 Place-and-Route tools (PAR) .....	58
5.3 CASTLE ve Ardışık Düzen Yapısı İle Gerçeklenen HSA Simülatörü .....	58
5.3.1 Orijinal CASTLE Mimarisi .....	59
5.3.2 Ardışık (Pipe-line) yapılı CASTLE Mimarisi .....	60
5.3.3 Bloklar içinde Ardışıklık (Pipelining) .....	61
5.3.4 Kayıtçı Grubuna Giriş Nasıl Sağlanır? .....	63
5.3.5 SONUÇ .....	63

## BÖLÜM 6

İKİLİK SAYI SİSTEMİ VE SABİT NOKTALI ARİTMETİK .....	65
6.1. İkilik Sayı Sisteminde İfade Edilen Bir Tamsayının Onluk Sistemdeki Karşılığı .....	65
6.2 İkilik Sistemde İfade Edilen Bir Sayının Bir Tümleyeni .....	65
6.3 İkilik Sistemde İfade Edilen Bir Sayının İkiye Tümleyeni .....	66
6.4 Negatif Bir Sayının İkilik Sistemdeki Gösterimi .....	66
6.5 Ondalık Sayıların İkilik Sayı Sisteminde Gösterimleri .....	66
6.6 Sekiz Bitlik Sayılarla Matematiksel İşlemler .....	69
6.6.1. Q1.7 Gösterimi .....	69
6.6.2. Q2.6 Gösterimi .....	69
6.6.3. $Q1.7 \times Q2.6 = Q3.13$ Çarpımı .....	69
6.6.4. $Q1.7 + Q2.6 = Q2.6$ Toplamı .....	70
6.6.5. Gerçek Sayılar Kullanılan Bir Örnek .....	71

## BÖLÜM 7

2 BOYUTLU HSA GABOR SÜZGECİN MATLAB VE MODELSİM BENZETİMLERİ .....	72
7.1 MATLAB Benzetimi .....	75
7.1.1. HSA Benzetimi .....	75
7.1.2. Kayan Noktalı Sayılar ile Benzetim .....	77
7.1.3 Sabit Noktalı Aritmetik ile Benzetim .....	80
7.2 Modelsim Benzetimi .....	83

7.2.1 HSA Gabor Filtresinin Modelsim Benzetimi .....	85
7.3 HSA Gabor Tip Süzgecin FPGA Gerçeklemesi .....	91
BÖLÜM 8	
SONUÇLAR .....	80
KAYNAKLAR .....	82
ÖZGEÇMİŞ .....	84

## SİMGELER

$u_{ij}, x_{ij}, y_{ij}$	HSA yapısının giriş, durum ve çıkışları
$ij$	ızgara yapısı düğüm tanımlayıcı
$r$	HSA komşuluk yarıçapı
$A(i, k; k, l)$	doğrusal geribesleme şablonu
$B(i, k; k, l)$	doğrusal ileri besleme şablonu
$f(.)$	doğrusal olmayan çıkış fonksiyonu
$\dot{x}_{ij}, \dot{v}(.)$	zamanda türev
$R_x$	bir HSA hücresinin doğrusal direnci
$C$	bir HSA hücresinin doğrusal kapasitesi
$\tau$	bir HSA hücresinin zaman sabiti
$T$	sayısal adım zamanı
$a_{i,j}, b_{i,j}$	geri ve ileri besleme operatörlerinin ağırlıkları
$U(m, n), V(m, n)$	HSA Gabor tip süzgecin giriş ve çıkışları
$V_e(m, n), V_o(m, n)$	HSA Gabor tip süzgecin gerçek ve sanal kısımları
$\omega_x, \omega_y$	uzamsal frekans
$\theta_o$	Gabor süzgecin yön açısı
$F(e^{j\omega_x}, e^{j\omega_y})$	iki boyutlu ayrık Fourier dönüşümü

## ŞEKİLLER LİSTESİ

2.1. Resim piksel yapısı .....	5
2.2. (a) $r=1$ ( $3 \times 3$ komşuluk),(b) $r=2$ ( $5 \times 5$ komşuluk) .....	5
2.3. Sınır ve köşe hücreler .....	6
2.4. Lineer olmayan çıkış .....	7
2.5. Sabit (Dirichlet) Sınır Koşullarının devre gösterimi .....	10
2.6. Sıfır akı (Neumann) sınır koşulları devre gösterimi .....	11
2.7. Periyodik (Toroidal) sınır koşulları devre gösterimi .....	11
2.8. Üç farklı paketleme yöntemi .....	12
2.9. $\tilde{A}$ ve $\tilde{B}$ nin bant yapısı .....	13
2.10. $3 \times 3 S_1(i, j)$ komşuluklu uzayda değişmeyen bir HSA. (a) $3 \times 3$ komşuluklu HSA'nın işaret akış diyagramı. (b) $C(i, j)$ hücresinin sistem yapısı. Koyu tonda basılmış oklar, giriş ve çıkışın çevre hücrelerinden $u_{kl}$ ve $y_{kl}$ gelen veri yollarıdır. İnce basılmış oklar ise sırasıyla $z$ eşik, $u_{ij}$ giriş, $x_{ij}$ durum ve $y_{ij}$ çıkıştır .....	17
2.11. Sıfır geri besleme (ileri besleme) $HSA \in C(0, B, z)$ . (a) $3 \times 3$ komşuluklu sıfır geri besleme HSA'nın işaret akış diyagramı. (b) $C(i, j)$ hücresinin sistem yapısı. Koyu tonda basılmış ok çevre hücrelerden gelen giriş sinyalini gösterir. Bu durumda, kendine geri besleme ve çevre hücrelerin çıkışlarındaki kuplaj yoktur .....	18
2.12. Sıfır giriş (Autonomous) $HSA \in C(A, 0, z)$ . (a) $3 \times 3$ komşuluklu sıfır giriş HSA'nın işaret akış diyagramı. (b) $C(i, j)$ hücresinin sistem yapısı. Koyu tonda basılmış ok çevre hücrelerin çıkışlarından gelen geri beslemeyi gösterir. Bu durumda hiç giriş işareti yoktur .....	18
2.13. Ayrılmamış $HSA \in C(A^0, B, z)$ . (a) $3 \times 3$ komşuluklu ayrılmamış giriş HSA'nın işaret akış diyagramı. (b) $C(i, j)$ hücresinin sistem yapısı. Koyu tonda basılmış oklar çevre hücrelerden gelen giriş işaretini gösterir. Bu durumda yalnızca kendi kendine geri besleme mevcuttur, çevre hücre çıkışlarında kuplaş meydana gelememektedir .....	19
3.1. Gabor süzgecinin gerçek ve sanal kısmı .....	22
3.2. 2-D Gabor Süzgecinin Yön seçiciliği .....	23
3.3. Üst sırada çift Gabor, alt sırada tek Gabor .....	24
3.4. Tek boyutlu direnç ızgara yapılı HSA alçak geçiren süzgeç, (b) dürtü ve (c) frekans cevabı .....	28
3.5. İki hücreli tek boyutlu HSA Gabor süzgecin gerçekleşmesi .....	28
3.6. HSA alçak geçiren süzgecin lineer direnç ızgara yapılı devre gerçekleştirilmesi .....	33
3.7. HSA alçak geçiren süzgecin $\lambda = 0.4$ için frekans yanıtı .....	35
3.8. HSA Gabor tip Süzgecin frekans yanıtı .....	43

4.1. 2 boyutlu HSA Gabor tip süzgecin OTA ve direnç kullanılarak analog gerçeklenmesi .....	44
4.2. Sayısal 2 boyutlu HSA Gabor süzgecin akış diyagramı .....	50
4.3. 2 boyutlu HSA Gabor süzgeç işlem elemanı .....	51
5.1. FPGA'in genel yapısı .....	53
5.2. FPGA tasarım çevriminin genel bir görünüşü .....	54
5.3. VHDL temel yapısı .....	55
5.4. VHDL yapısal tanımlama örneği .....	56
5.5. Özel veya fonksiyonunun VHDL modeli .....	57
5.6. D tipi flip-flop .....	57
5.7. CASTLE mimarisi .....	59
5.8. Orijinal CASTLE aritmetik çekirdeği .....	60
5.9. Ardışık düzenli CASTLE mimarisi .....	61
5.10. Ardışık düzenli CASTLE mimarisinde hafıza kullanımı (1) .....	61
5.11. Ardışık düzenli CASTLE mimarisinde hafıza kullanımı (2) .....	61
5.12. Ardışık düzenli topayıcı .....	62
5.13. Ardışık düzenli "Baugh-Wooley" çarpıcısı .....	62
7.1. (a)121 Girişli FIR Gabor Yapısı , (b) HSA Gabor Tip Süzgeç birim işlem elemanı	73
7.2. (a) 31 uzunluklu bir sonlu süreli tek boyutlu alçak geçiren süzgecin frekans yanıtı, (b-d) (a)'da elde edilen süzgecin iki boyuta genişletilmiş durumunun dürtü yanıtı, (c) (b)'deki süzgecin frekans yanıtı .....	74
7.3. İntegralin dikdörtgen yaklaşıklığı ile hesaplanması .....	75
7.4. İki boyutlu HSA Gabor Süzgecin Sayısal devre gerçekleştirilmesi .....	78
7.5. (a) benzetim için kullanılan 128x128 boyutundaki giriş resmi. (b) ve (c) sırası ile çift ve tek gaborun görüntüleri .....	79
7.6. Sırası ile çift ve tek gaborun dürtü yanıtları .....	79
7.7. Sırası ile çift ve tek gaborun frekans yanıtları .....	80
7.8. 1.15 formatında $w_x=\pi/4$ , $w_y=\pi/4$ , $\lambda=0.5$ değerleri için (a) tek Gabor'un birim dürtü, (b) tek Gabor'un frekans, (c) çift Gabor'un birim dürtü, (d) çift Gabor'un frekans yanıtları .....	82
7.9. 1.7 formatında $w_x=\pi/4$ , $w_y=\pi/4$ , $\lambda=0.5$ değerleri için (a) tek Gabor'un birim dürtü, (b) tek Gabor'un frekans, (c) çift Gabor'un birim dürtü, (d) çift Gabor'un frekans yanıtları .....	82
7.10. 1.15 formatında $w_x=\pi/4$ , $w_y=\pi/4$ , $\lambda=0.2$ değerleri için (a) tek Gabor'un birim dürtü, (b) tek Gabor'un frekans, (c) çift Gabor'un birim dürtü, (d) çift Gabor'un frekans yanıtları .....	82
7.11. 1.7 formatında $w_x=\pi/4$ , $w_y=\pi/4$ , $\lambda=0.2$ değerleri için (a) tek Gabor'un birim dürtü, (b) tek Gabor'un frekans, (c) çift Gabor'un birim dürtü, (d) çift Gabor'un frekans yanıtları .....	82
7.12: Modelsim giriş sayfası .....	83
7.13: Modelsim içinde yazılmış VHDL kodu .....	84
7.14: Modelsim benzetim sonuçları .....	84
7.15: Genel yapı .....	85
7.16: Satır satır tarama işlemi .....	86
7.17: Giriş ve durum kayıtçı tabloları .....	86
7.18: İşaret akışı .....	87
7.19: İşaret durumları .....	88
7.20: Modelsim benzetim giriş ve çıkış sayısal görüntüleri. Açık değerleri x yönünde $45^0$ , y yönünde ise $-45^0$ , bant genişliği ise 0.1 alınmıştır .....	89
7.21: Gabor Filtresi FPGA benzetimi genel yapısı .....	90

## TABLolar LİSTESİ

5.1: Farklı gecikmeler .....	60
5.2 : Gecikme zamanları ve ip buyklkleri .....	64

## ÖNSÖZ

Tezin hazırlanması sırasında bana her türlü desteđi veren danışman hocam sayın Prof. Dr. Vedat TAVŞANOĐLU`na, her zaman bana destek olan aileme, dayım Metin ERDOĐDU' ya, program, görüş ve önerileri için en yakın arkadaşlarım Seyhan AĐAOĐLU ve Olcay SEVİM'e en sonda ise her şeyi paylaştığım Burçin ÖĐRETEN' e teşekkürlerimi sunarım.



## ÖZET

Hücrel Ağ modeli 1988 yılında L. O. Chua ve L. Yang tarafından ortaya atılmıştır. Hücrel Sinir Ağlarının, insan beynindeki gibi bölgesel bağlantılara sahip olması VLSI gerçeklenmesini olanaklı kılmıştır. Analog yapıdaki HSA, sayısal sistemlerin erişemeyeceği işlem hızlarına erişebilmekte, bundan dolayı birçok alanda tercih edilmektedir. HSA komşuluk ilişkileri bulunan düzenli bir ızgara yapısına sahiptir. Hücrel Sinir Ağlarının sahip olduğu bu yapı birçok uygulamaya elverişli olsa da en yaygın kullanımını görüntü işlemede bulmaktadır. HSA ile görüntü işleme için çeşitli filtreler kullanılmaktadır, bunlardan biri de GABOR Filtreleridir.

Gabor süzgeçleri görüntü işlemede ön işlem olarak kullanılırlar. Gabor Süzgecinin Sayısal devrelerde gerçeklenmesinde karmaşık sayılar yüzünden bazı problemler oluşmaktadır. Bu tezde gabor süzgeçlerinin sayısal ortamda daha iyi nasıl gerçeklenebileceği ve bu gerçeklemenin uygulamaları sunulacaktır. Gabor süzgecinin karmaşık işlem yoğunluğunu HSA (Hücrel Sinir Ağ) ile nasıl azaltılabileceğini gösteren B. Shi HSA Gabor Tip Süzgeçleri geliştirmiştir. Analog VLSI olarak gerçeklenen bu süzgeçler hem gerçek hem de sanal kısımlarını aynı anda hesaplama ve analog bir sistem olması nedeniyle yüksek hızlar sunmaktadır. Bu tezde Analog sistemin yaklaşıklık yöntemleri kullanılarak sayısal devre durumuna getirilip sayısal olarak HSA Gabor Tip Süzgecin MATLAB ve Modelsim benzetimi ve FPGA (Field Programming Gate Array) üzerinde gerçeklenmesi yapılmıştır.

**Anahtar kelimeler:** Hücrel Sinir Ağ (HSA), Gabor, süzgeç, FPGA, VHDL.

## **ABSTRACT**

Cellular neural network model was first introduced in 1988 by L.O. Chua and L. Yang. CNN has local connections as Human's brain has and this important makes VLSI implementation possible. Analog constructed CNN has powerful speeds compared to digital systems. Because of this, CNN is preferred mostly for different purposes. CNN has regular grid interconnections that have neighbourhood. The mentioned construction is convenient for most applications but the mostly chosen area is image processing CNN uses some filters for image processing and one of them is Gabor filters.

Gabor filters are used for pre processing in image processing. When implementing Gabor filters on digital system, we face some problems due to complex manipulation. I describe how to implement Gabor filter on digital systems with improvements, and show applications about these implementations in my thesis. B. Shi has developed CNN Gabor filters to reduce complex computation. This system is built analog VLSI and also has the capability of computing both real parts and imaginary parts at the same time which lead the system to an incredible speed of processing. I used some approximation methodologies to convert analog system to digital system in manner of implementation, then I perform CNN Gabor filtering digitally in MATLAB and Modelsim. In addition to all, I implement this algorithm on FPGA.

**Key words:** Cellular Neural Network (CNN), Gabor, filter, VHDL, FPGA

## 1. GİRİŞ

Elektronik ve bilgisayar sektörünün yakın tarihine bakıldığında üç devrim dalgası görülür. İlk devrim 1970'lerden 1980'lerdeki PC (personal computer) endüstrisine kadar süren mikroişlemciler ile geliştirilen endüstriyel bilgisayarlar devrimi. İkincisi ucuz lazer ve fiber optik nedeniyle 1980'lerin sonundan 1990'lardaki internet endüstrisine neden olan geniş bant. Üçüncü devrim dalgası ise 1990'ların sonundaki duyar endüstrisidir. Duyar endüstrisinin anlamı ucuz duyar ve MEMS (micro-electro-mechanical system) dizilerinin makul formlar için çoğalmasdır. Yapay gözler, burun, kulaklar, tat alma gibi insan ya da hayvan kullanılmayacak kimyasal, fiziki ve biyolojik endüstride kullanılmaları. İnsan hatta hayvanlar için kaybettikleri duyularının tekrar yerine konması gibi. Bu yapılarda süreç içinde işlenmeyi bekleyen milyonlarca analog işaret vardır. Sonuç olarak yeni bir hesaplama modeline ihtiyaç vardır.

Analog hesaplamaları yapmak için Hücresel Bilgisayar (Analogic Cellular Computer) modeli iyi bir adaydır. Bu bilgisayarın çekirdeği analog dinamik hücreler dizisinden oluşan Hücresel Sinir Ağı yapısıdır. Bu bilgisayar mimarisi CNN Universal Machine (CNN-UM) dir ve bir çok fiziksel uygulaması geliştirilmiştir.

Hücresel Ağ modeli 1988 yılında L. O. Chua ve L. Yang tarafından ortaya atılmıştır. Hücresel Sinir Ağlarının, insan beynindeki gibi bölgesel bağlantılara sahip olması VLSI gerçekleşmesini olanaklı kılmıştır. Analog yapıdaki HSA sayısal sistemlerin erişemeyeceği işlem hızlarına erişebilmekte, bundan dolayı birçok alanda tercih edilmektedir [1].

Bilgi alış-verişinde görsel verinin rolü oldukça fazladır. Gelişen teknoloji ile birlikte görüntü ve resim gibi görsel veriye artan talep, daha çok verinin işleme gerekliliğini de beraberinde getirmiştir.

Hücresel Sinir Ağlarının sahip olduğu yapı birçok uygulamaya elverişli olsa da en yaygın kullanımını görüntü işlemede bulmaktadır.

Başta da söylediğim gibi yapay organ, görüntü ve resim işleme gibi süreçler için yoğun hesaplamalar gerekmektedir. Hız konusuna bir örnek vermek gerekirse; dakikada 30 pencerenin değiştiği 512 x 512'lik bir görüntü, 3 x 3'lük bir maske ile konvolüsyon edilmek istense, dakikada 70.7 milyon çarpma ve 62.9 milyon toplama işlemi yapılması gerekmektedir. Bu konvolüsyonun sayısal bir sistemle yapılması saniyeler alırken, HSA kullanıldığında görüntünün boyutundan bağımsız olarak işlem nano ( $10^{-9}$ ) saniyeler sonunda tamamlanmış olacaktır.

Gabor süzgeçleri bilgisayarla görme ve görüntü işleme konularındaki bir uygulama için farklı kullanımları vardır. Bunlara örnek olarak doku tanıma ve sınıflama [2], yazı tanımlama [3], doku ayırımı [4], kenar belirleme [5], görüntü sıkıştırma [6], hareket kestirimi [7], nesne tanımlama [8] ve dokuda şekil tanıma [9]. Gabor süzgecinin yön seçici özelliği kullanılarak gerçekleştirilen görsel korteks [10], bu yöntemlerin bilgisayarla görme ve sayısal görüntü işlemede kullanımını arttırmıştır. Marceljia [10] Gabor süzgecinin memelilerin görsel sistemi için kullanılabileceğini ilk öne süren kişidir. Üç boyutlu (3-D) Gabor süzgeçleri, cortical hücrelerinin hız ve yön hassasiyetinin modellenmesinde kullanılmıştır. Çünkü bir görüntüdeki hareket, zaman uzayında yer değişimi gösteren bir “zaman vektörü” olarak düşünülebilir. Gabor süzgecinin tek eksik yanı fazla hesaplama gerektirmesidir.

Yoğun hesaplamalardan kurtulmak için HSA analog yapısı kullanılarak HSA Gabor Tip Süzgeç oluşturuldu [10]. Süzgeç katsayıları karmaşık değerlidir. Gabor süzgecini görüntülere uygulamak için ayrı Gabor fonksiyonunun bulunması gerekmektedir. Gabor süzgecinin hesaplama yükü ayrı düzlemdeki konvolüsyon gerekliliğini engellemektedir. Yakın bir zamanda, bu problem Shi tarafından çözülmüştür. Shi Gabor süzgecine benzeyen konvolüsyon şablonlarını bularak görüntü üzerinde uygulamıştır.

Bu tez aşağıdaki bölümlerden oluşmaktadır. Bölüm 2'de Hücrel Sinir Ağı nedir, matematiksel modeli nasıldır gibi soruların cevapları verilecektir. Gabor süzgeçleri, neden HSA Gabor tip süzgeç yapısı gibi sorulara ise Bölüm 3'de cevaplandırılacaktır. Bölüm 4'te iki boyutlu HSA Gabor süzgecin analog ve sayısal tasarımları verilmektedir. Sayısal gerçekleştirme için donanım seçimi, FPGA donanım ve sayısal yapısı, HSA için geliştirilen FPGA algoritması ve gerçekleştirilmesi bölüm 5'te verilmektedir. Sayısal gerçekleştirilmede kullanılacak temel aritmetik ve sayı sistemleri bölüm 6'nın konusunu oluşturmaktadır. Bölüm 7 ve 8 ise verilen sayısal yaklaşım ve modellere göre iki boyutlu HSA Gabor tip süzgecin

benzetimi (MATLAB ve MODELSİM) ve devre gereklemesi verilecektir. Son blümde ise sonuçlar sunulmuştur.

## 2. HÜCRESEL SINİR AĞLARI

Hopfield Sinir Ağı'nın özel bir kümesi olan Hücresel Sinir Ağları 1988 yılında ortaya atılmıştır. Bu bölümde Hücresel Sinir Ağları kavramı ve temel tanımlamaları verilecektir.

Sinir sistemimizdeki nöronların birbirine olan bağlantılarına yoğunlaşan iki bilim adamı Leon O. Chua ve Lin Yang 1988 yılında yeni bir yapay sinir ağı mimarisi ortaya koydukları makalelerini yayınladılar ve bu makale ile Hücresel Sinir Ağı (HSA) doğmuş oldu [1].

Sinir sistemimizdeki nöronların diğer nöronlarla bölgesel olarak bağlı olması -ki her bir nöronun 1000'e yakın komşuluk kurduğu nöron bulunduğu bilinmektedir- HSA yapısının dayandığı en önemli özelliktir. Hopfield sinir ağlarındaki bütün hücrelerin birbirleriyle doğrudan bağlantısına karşın HSA'nda her bir hücre en yakın komşuluğundaki hücre ile doğrudan, diğer hücrelerle dolaylı olarak bağlıdır. HSA'nın bu yapısı, ağın tümleşik devre gerçekleştirmesini olanaklı kılmış ve ilk HSA çipi (CNN-UM) Macaristan Teknik Üniversitesi'nde Prof. T. Roska ve öğrencileri tarafından gerçekleştirilmiştir.

Bölgesel bağlantı sonucu; hem devrenin kapladığı alan azalmakta hem de verilerin devre üzerindeki taşınım yolları kısa olduğundan toplam güç harcaması düşük olmaktadır.

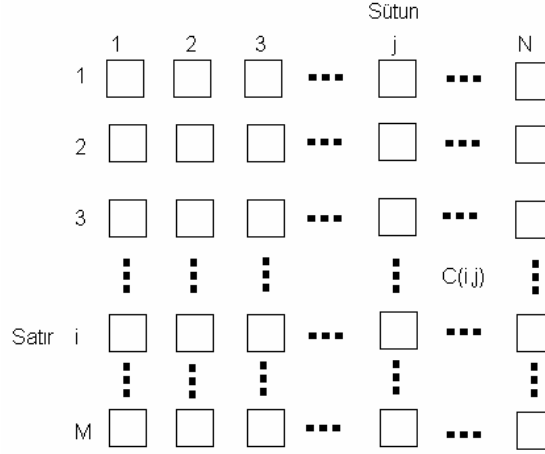
HSA önemli avantajlarından bir diğeri de kararlı bir sonuca yakınsama süresinin devrenin boyutundan bağımsız olması ve bir kaç nöron gecikmesi ile belirlenmesidir.

HSA'nın bu özellikleri gelecekte pek çok uygulama alanı bulacağına göstergeleridir.

## 2.1. Temel Gösterim ve Tanımlamalar

### 2.1.1. Standart HSA Mimarisi

Standart HSA mimarisi  $M \times N$  boyutundaki hücreler  $(C(i, j))$  dizisinden oluşmaktadır.  $(i, j), i = 1, 2, \dots, M, j = 1, 2, \dots, N$  kartezyen koordinatları gösterir (şekil 2.1).

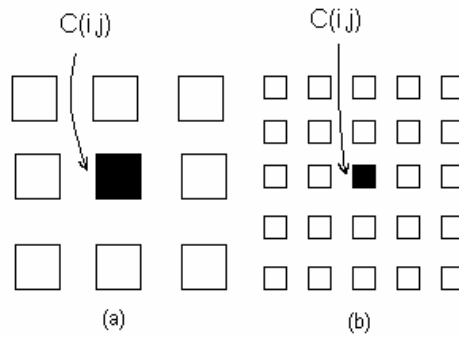


Şekil 2.1 : Resim piksel yapısı

### 2.1.2. $(C(i, j))$ Hücresinin Etki Küresi

$C(i, j)$  hücresinin  $r$  yarıçaplı  $S_r(i, j)$  etki küresi şekil 2.2'de belirtilen durumdaki bütün komşularının oluşturduğu bir kümedir.

$$S_r(i, j) = \{C(k, l) \mid 1 \leq k \leq M, 1 \leq l \leq N, \max\{|k - i|, |l - j|\} \leq r\}$$



Şekil 2.2 (a)  $r=1$ (3x3 komşuluk), (b)  $r=2$ (5x5 komşuluk)

Bazı durumlarda  $S_r(i, j)$  yerine  $(2r+1) \times (2r+1)$  komşuluğu olarak kullanılır. Şekil 2.2-a' da  $r=1(3 \times 3)$  komşuluk, şekil 2.2-b' de ise  $r=2(5 \times 5)$  komşuluk gösterilmektedir.

### 2.1.3. Normal ve Sınır Hücreleri

Eğer  $C(i, j)$  hücresinin  $S_r(i, j)$  ile gösterilen bütün komşuları  $C(k, l) \in S_r(i, j)$  durumunu sağlarsa bu  $C(i, j)$  hücresine normal hücre denir. Diğer durumdaki  $C(i, j)$  hücrelerine ise sınır hücreleri denir (şekil 2.3).

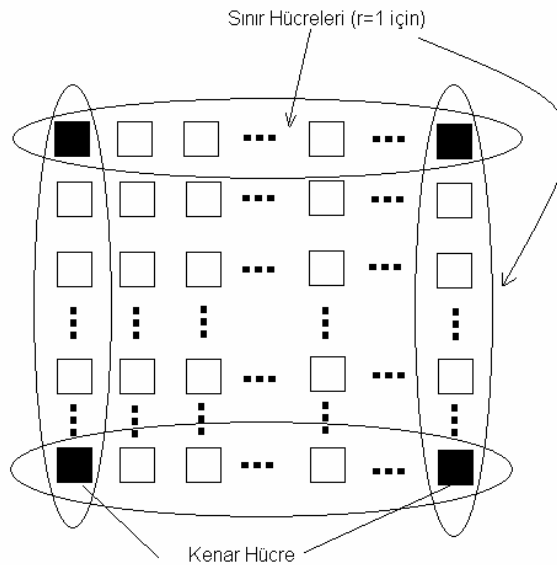
En dıştaki sınır hücrelerine kenar hücreleri de denir, Eğer  $r > 1$  ise bütün sınır hücreleri kenar hücre olmaz.

### 2.1.4. Standart HSA

$M \times N$  boyutundaki standart bir HSA  $(i, j), i=1, 2, \dots, M, j=1, 2, \dots, N$  düzlemine yerleşmiş  $M \times N$  boyutundaki dikdörtgen  $C(i, j)$  hücreler dizisidir.  $C(i, j)$  hücresinin matematiksel bağıntısı:

Durum denklemi

$$\dot{x}_{ij} = -x_{ij} + \sum_{C(k,l) \in S_r(i,j)} A(i, j; k, l) y_{kl} + \sum_{C(k,l) \in S_r(i,j)} B(i, j; k, l) u_{kl} + z_{ij} \quad (2.1)$$



Şekil 2.3 : Sınır ve köşe hücreler

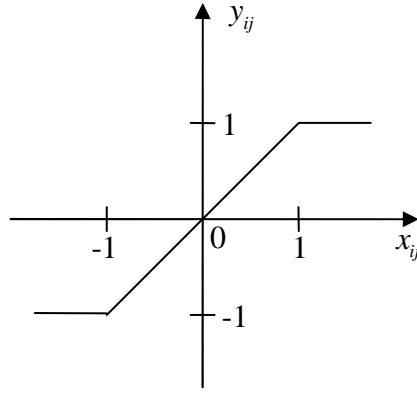


$x_{ij} \in \square, y_{kl} \in \square, u_{kl} \in \square$  ve  $z_{ij} \in \square$  sırası ile  $C(i, j)$  hücresinin durumu, çıkışı, girişi ve eşiği olarak adlandırılır.  $A(i, j; k, l)$  ve  $B(i, j; k, l)$  ise geri besleme ve giriş sinaptik operatörlerdir.

### Çıkış Denklemi

$$y_{ij} = f(x_{ij}) = \frac{1}{2} |x_{ij} + 1| - \frac{1}{2} |x_{ij} - 1| \quad (2.2)$$

Şeklinde tanımlanan doğrusal olmayan bir denklemdir (şekil 2.4).



Şekil 2.4 : Lineer olmayan çıkış

### Sınır Koşulları:

Çıkış  $y_{kl}$  ve giriş  $u_{kl}$  dizilerinde, hücre etki alanının boyutu oranında bir fazlalık meydana gelir. Konvolüsyon sonucunda gelen fazlalık sorununa 2.2.1.2 de değinilmektedir.

### İlk Durum;

$$x_{ij}(0), i = 1, \dots, M, j = 1, \dots, N$$

### Notlar:

1.  $u_{kl}$  giriş değeri  $M \times N$  boyutundaki gri ölçeklenmiş P görüntüsünün piksel yeğinliğidir. Normalize edilen P resminin piksel değerleri  $-1 \leq u_{kl} \leq 1$  arasındadır ve -1 beyaz, +1 ise siyah olarak kodlanmıştır. Durağan resimler için  $u_{kl}$  zamanda sabittir, hareketli resimlerde

(video) ise  $u_{kl}$  zamanın bir fonksiyonudur. Diğer değişkenler ise  $(x(0), y, z)$  her resim için özel olarak belirlenir.

2. Genel olarak  $A(i, j; k, l)$ ,  $B(i, j; k, l)$  ve  $z_{ij}$  değerleri uzayda  $(i, j)$  ve zamanda  $t$  değişebilir. Bunun dışındaki durumlara zamanda değişmezlik veya uzayda değişmezlik denir.

3.  $x_{kl}(t), y_{kl}(t), u_{kl}(t), x_{ij}(t), y_{ij}(t)$  ve  $u_{kl}(t)$  ( $0 \leq t \leq t_0$ ) üzerinde işlem yapan  $A(i, j; k, l)$  ve  $B(i, j; k, l)$  operatörleri çoğu zaman lineer değildir.

### 2.1.5. Uzayda Değişmeyen ya da Eşyönlü HSA

Eğer  $A(i, j; k, l), B(i, j; k, l)$  ve  $z_{ij}$  uzayda değişmiyorsa bu HSA ya uzayda değişmeyen ya da eşyönlü HSA denir.

$$\begin{aligned} \sum_{C(k,l) \in S_r(i,j)} A(i, j; k, l) y_{kl} &= \sum_{|k-i| \leq r} \sum_{|l-j| \leq r} A(i-k; j-l) y_{kl} \\ \sum_{C(k,l) \in S_r(i,j)} B(i, j; k, l) u_{kl} &= \sum_{|k-i| \leq r} \sum_{|l-j| \leq r} B(i-k; j-l) u_{kl} \\ z_{ij} &= z \end{aligned}$$

Standart HSA (lineer sinaptik operatörler) aşağıdaki durum denkleminde sahiptir.

$$\begin{aligned} \dot{x}_{ij} &= -x_{ij} + \sum_{C(k,l) \in S_r(i,j)} A(i, j; k, l) y_{kl} + \sum_{C(k,l) \in S_r(i,j)} B(i, j; k, l) u_{kl} + z_{ij} \\ &+ \sum_{C(k,l) \in S_r(i,j)} C(i, j; k, l) x_{kl} + \sum_{C(k,l) \in S_r(i,j)} D(i, j; k, l) (u_{kl}, x_{kl}, y_{kl}) \end{aligned} \quad (2.3)$$

## 2.2 Matematiksel Temeller

### 2.2.1 Vektör, Matris Gösterimleri ve Sınır Koşulları

$n = MN$  fark denklemleri sistemi olarak tanımlanan standart HSA (2.4) gösterilmiştir.

$$\dot{x}_{ij} = h_{ij}(\tilde{\chi}_{ij}, t), \quad i = 1, 2, \dots, M, \quad j = 1, 2, \dots, N \quad (2.4)$$

$$h_{ij}(\tilde{\mathcal{X}}_{ij}, t) = -x_{ij}(t) + \sum_{C(k,l) \in S_r(i,j)} A(i, j; k, l) y_{kl}(t) + s_{ij}(t)$$

$$y_{kl} = f(x_{kl})$$

$$s_{ij}(t) = \sum_{C(k,l) \in S_r(i,j)} B(i, j; k, l) u_{kl}(t) + z_{ij}(t)$$

$\tilde{\mathcal{X}}_{ij}$ ,  $x_{kl} \in S_r(i, j)$  değerlerini içeren  $(2r+1)^2$  boyutunda bir vektördür.

$$\{x_{kl} : |k-i| \leq r, |l-j| \leq r\}$$

(2.6) denklemine birebir uyan HSA mimarisi  $M \times N$  matris fark denklemi şeklinde gösterilebilir.

$$\begin{bmatrix} \dot{x}_{11} & \dot{x}_{12} & \dots & \dot{x}_{1N} \\ \dot{x}_{21} & \dot{x}_{22} & \dots & \dot{x}_{2N} \\ \vdots & \vdots & & \vdots \\ \dot{x}_{M-1,1} & \dot{x}_{M-1,2} & \dots & \dot{x}_{M-1,N} \\ \dot{x}_{M1} & \dot{x}_{M1} & \dots & \dot{x}_{MN} \end{bmatrix} = \begin{bmatrix} h_{11}(\tilde{\mathcal{X}}_{11}) & h_{12}(\tilde{\mathcal{X}}_{12}) & \dots & h_{1N}(\tilde{\mathcal{X}}_{1N}) \\ h_{21}(\tilde{\mathcal{X}}_{21}) & h_{22}(\tilde{\mathcal{X}}_{22}) & \dots & h_{2N}(\tilde{\mathcal{X}}_{2N}) \\ \vdots & \vdots & & \vdots \\ h_{M-1,1}(\tilde{\mathcal{X}}_{M-1,1}) & h_{M-1,2}(\tilde{\mathcal{X}}_{M-1,2}) & \dots & h_{M-1,N}(\tilde{\mathcal{X}}_{M-1,N}) \\ h_{M1}(\tilde{\mathcal{X}}_{M1}) & h_{M2}(\tilde{\mathcal{X}}_{M2}) & \dots & h_{MN}(\tilde{\mathcal{X}}_{MN}) \end{bmatrix} \quad (2.5)$$

### 2.2.1.1. Sanal hücreler

$|k-i| \leq r, |l-j| \leq r; k \notin \{1, 2, \dots, M\}$  ve/veya  $l \notin \{1, 2, \dots, N\}$  koşulunu sağlayan herhangi bir  $C(k, l)$  hücresine sanal hücre denir. Ve buna ilişkin  $x_{kl}, y_{kl}, u_{kl}$  ve  $z_{kl}$  sırasıyla sanal durum, sanal çıkış, sanal giriş ve sanal eşik olarak adlandırılır.

### 2.2.1.2. Sınır Koşulları

Eşitlik (2.4) in  $\tilde{\mathcal{X}}_{ij}$ indeki sanal değerler sınır koşulları ile belirlenmektedir.

#### 1 Sabit (Dirichlet) Sınır Koşulları

$$\text{Sol sanal hücreler: } y_{i,0} = \alpha_1, \quad u_{i,0} = \beta_1, \quad i = 1, 2, \dots, M$$

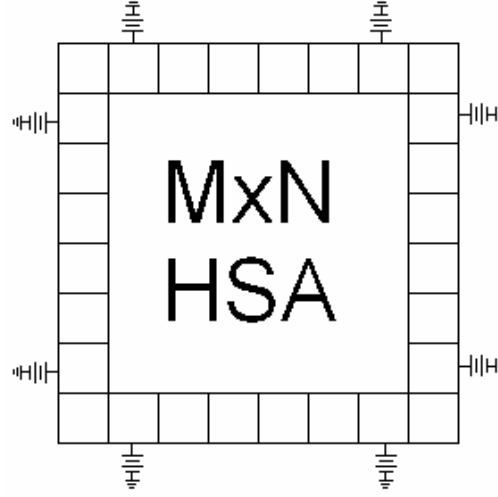
$$\text{Sağ sanal hücreler: } y_{i,N+1} = \alpha_2, \quad u_{i,N+1} = \beta_2, \quad i = 1, 2, \dots, M$$

$$\text{Üst sanal hücreler: } y_{0,j} = \alpha_3, \quad u_{0,j} = \beta_3, \quad j = 1, 2, \dots, N$$

$$\text{Alt sanal hücreler: } y_{M+1,j} = \alpha_4, \quad u_{M+1,j} = \beta_4, \quad j = 1, 2, \dots, N$$

$\alpha_i$  ve  $\beta_i$  değerleri birer sabit olup genellikle sıfır alınır.

Devre yorumu: Sınırlara birer sütun ve satır eklenir, eklenen bu satırlardaki hücelere bir kaynak vasıtasıyla sabit değerler atanır (şekil 2.5).



Şekil 2.5 Sabit (Dirichlet) Sınır Koşullarının devre gösterimi

2 Sıfır akı (Neumann) sınır koşulları

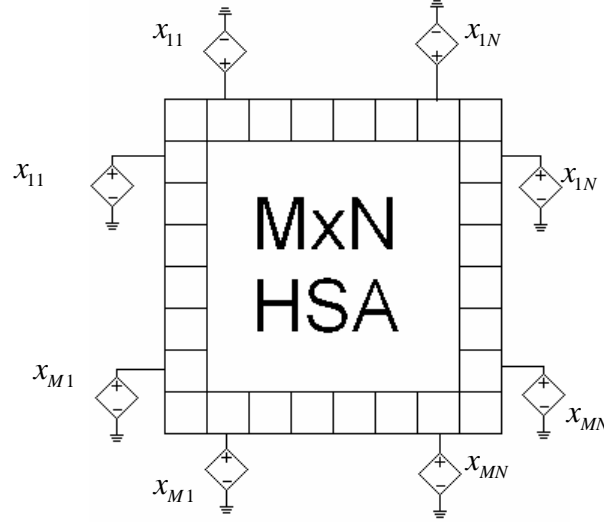
$$\text{Sol sanal hücreler: } y_{i,0} = y_{i,1}, \quad u_{i,0} = u_{i,1}, \quad i = 1, 2, \dots, M$$

$$\text{Sağ sanal hücreler: } y_{i,N+1} = y_{i,N}, \quad u_{i,N+1} = u_{i,N}, \quad i = 1, 2, \dots, M$$

$$\text{Üst sanal hücreler: } y_{0,j} = y_{1,j}, \quad u_{0,j} = u_{1,j}, \quad j = 1, 2, \dots, N$$

$$\text{Alt sanal hücreler: } y_{M+1,j} = y_{M,j}, \quad u_{M+1,j} = u_{M,j}, \quad j = 1, 2, \dots, N$$

Genellikle girişin sıfır ya da olmadığı zamanlarda uygulanır. Çünkü herhangi bir giriş dışarıdan madde veya enerji girişine neden olur, buda sistemi açık sistem yapar. Sıfır girişli HSA ya özerk HSA olarak adlandırılır.



Şekil 2.6 Sıfır akı (Neumann) sınır koşulları devre gösterimi

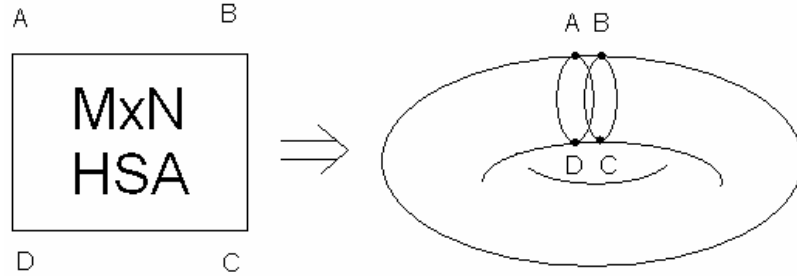
### 3 Periyodik (Toroidal) sınır koşulları

$$\text{Sol sanal hücreler: } y_{i,0} = y_{i,N}, \quad u_{i,0} = u_{i,N}, \quad i = 1, 2, \dots, M$$

$$\text{Sağ sanal hücreler: } y_{i,N+1} = y_{i,1}, \quad u_{i,N+1} = u_{i,1}, \quad i = 1, 2, \dots, M$$

$$\text{Üst sanal hücreler: } y_{0,j} = y_{M,j}, \quad u_{0,j} = u_{M,j}, \quad j = 1, 2, \dots, N$$

$$\text{Alt sanal hücreler: } y_{M+1,j} = y_{1,j}, \quad u_{M+1,j} = u_{1,j}, \quad j = 1, 2, \dots, N$$



Şekil 2.7 Periyodik (Toroidal) sınır koşulları devre gösterimi

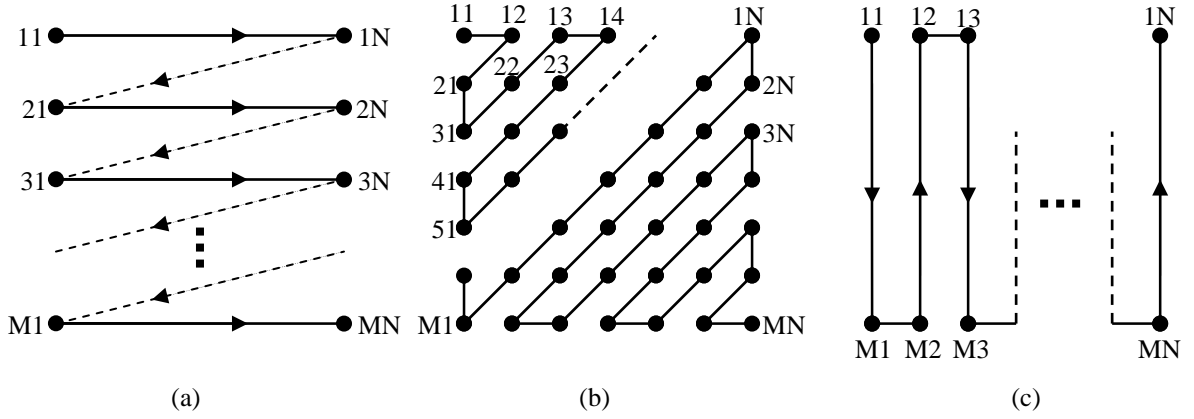
Üst satırdaki bütün hücreler en alt satırdaki hücreler ile tanımlanır. En sol satırdaki hücreler ise en sağdaki hücreler ile tanımlanır.

#### 2.2.1.3. Vektör fark denklemi

Neredeyse bütün fark denklemlerinin çözümü için kullanılan teorem ve sayısal tekniklerin hepsi vektör formatındadır, bu yüzden (2.4) da verilen  $M \times N$  matris fark denklemi  $MN \times 1$  boyutundaki vektör fark denklemine dönüştürülmesi gerekir. Dönüştürülme işlemi paketleme yöntemi ile yapılır. Üç tip paketleme yöntemi kullanılmaktadır.

1. Satır paketleme yöntemi (şekil 2.8-a)
- 2 Köşegen paketleme yöntemi (şekil 2.8-b)
- 3 Sütun paketleme yöntemi (şekil 2.8-c)

Bu paketleme yöntemleri sırasıyla Şekil 2.8 (a),(b) ve (c) de gösterildi.



Şekil 2.8 Üç farklı paketleme yöntemi

Paketlemeden sonra aşağıdaki  $n = MN$  vektör sistemi ve bu sistemin vektör formu bulunur.

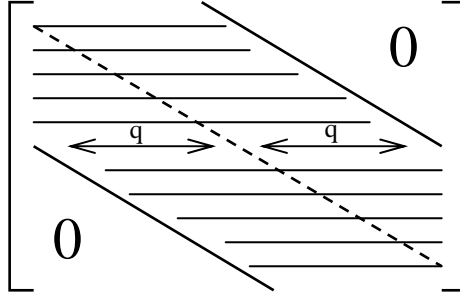
$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \vdots \\ \dot{\hat{x}}_n \end{bmatrix} = - \underbrace{\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_n \end{bmatrix}}_x + \underbrace{\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}}_y + \underbrace{\begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_n \end{bmatrix}}_u + \underbrace{\begin{bmatrix} \hat{z}_1 \\ \hat{z}_2 \\ \vdots \\ \hat{z}_n \end{bmatrix}}_z$$

$$\dot{x} = -x + \hat{A}y + \hat{B}u(t) + z(t)$$

$$y_i = f(x_i)$$

$$x = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n]^T$$

$\hat{A}$  ve  $\hat{B}$   $n \times n$  boyutundaki matrisler üç paketleme yöntemi ile elde edilebilen  $A(i, j; k, l)$  ve  $B(i, j; k, l)$  sinaptik ağırlıklarıdır. Her matris şekil 2.9 de görüldüğü gibi bant yapısındadır (çoğu giriş sıfırdır).



Şekil 2.9  $\tilde{A}$  ve  $\tilde{B}$  nin bant yapısı

$M = N$  için bant genişliği  $w = 2q + 1$  formülü ile bulunur.  $q$  değeri paketleme yöntemlerine göre değişir, bu değerlerin hesaplanması aşağıdaki formüllerle göre yapılır.

Satır paketleme yöntemi için  $q = N + 1$ ,

Köşegen paketleme yöntemi için  $q = 2n - 2$ ,

Sütun paketleme yöntemi için  $q = 2N - 1$ .

$\hat{A}$  ve  $\hat{B}$  çok büyük matrislerdir. HDTV uygulaması için  $M = N = 1000$  alınır,  $n = 1000000$ ,  $q = 1001$ ,  $w = 2003$  olur satır paketleme yöntemi için. Görüldüğü gibi matrisin toplam boyutunun 0,2% si kadar bir bant genişliği gerekli.

## 2.2.2 Uzayda Değişmeyen HSA

### 2.2.2.1. Klonlayan şablon gösterimi

HSA uygulamalarında  $3 \times 3$  komşuluklu uzayda değişmeyen standart HSA yapısı kullanılır, bundan dolayı daha sonraki analizler için bazı gösterim ve teknik terimler açıklanması son derece uygundur.

$C(i, j) \in S_r(i, j)$  hücresi göz önüne alınırsa;

$$\begin{pmatrix} C(i-1, j-1) & C(i-1, j) & C(i-1, j+1) \\ C(i, j-1) & C(i, j) & C(i, j+1) \\ C(i+1, j-1) & C(i+1, j) & C(i+1, j+1) \end{pmatrix}$$

- HSA denklemindeki iki sinaptik ağırlık ve eşik öğelerinden gelen katkı

1  $A(i, j; k, l)$  geri besleme operatörünün katkısı

$$\begin{aligned}
 \sum_{C(k,l) \in S_r(i,j)} A(i, j; k, l) y_{kl} &= \sum_{|k-i| \leq 1} \sum_{|l-j| \leq 1} A(i-k; j-l) y_{kl} \\
 &= a_{-1,-1} y_{i-1,j-1} + a_{-1,0} y_{i-1,j} + a_{-1,1} y_{i-1,j+1} \\
 &\quad + a_{0,-1} y_{i,j-1} + a_{0,0} y_{i,j} + a_{0,1} y_{i,j+1} \\
 &\quad + a_{1,-1} y_{i+1,j-1} + a_{1,0} y_{i+1,j} + a_{1,1} y_{i+1,j+1} \\
 &= \sum_{k=-1}^1 \sum_{l=-1}^1 a_{k,l} y_{i+k,j+l} \tag{2.6}
 \end{aligned}$$

$$\square \begin{pmatrix} a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ a_{0,-1} & a_{0,0} & a_{0,1} \\ a_{1,-1} & a_{1,0} & a_{1,1} \end{pmatrix} \otimes \begin{pmatrix} y_{-1,-1} & y_{-1,0} & y_{-1,1} \\ y_{0,-1} & y_{0,0} & y_{0,1} \\ y_{1,-1} & y_{1,0} & y_{1,1} \end{pmatrix} = A \otimes Y_{ij} \tag{2.7}$$

3x3 boyutundaki  $\mathbf{A}$  matrisine geri besleme klonlayan şablonu denir, ve “ $\otimes$ ” sembolü eleman eleman çarpımların toplamını verir. Bundan sonra bu işarete şablon nokta çarpımı denecektir. Ayırık matematikte bu işleme uzamsal konvolüsyon denir.  $Y_{ij}$  3x3 boyutundaki matris  $M \times N$  boyutundaki çıkış resmi üzerindeki her bir  $C(i, j)$  noktası için ayrı ayrı tanımlanır.

$a_{kl}$  elemanına  $\mathbf{A}$  geri besleme şablonunun merkez elemanı, ağırlığı ya da katsayısı denir. Bazı durumlarda  $\mathbf{A}$  şablonunun ayrıştırılmış şekli kullanılır.

$$\begin{aligned}
 A &= A^0 + \tilde{A} \\
 A^0 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & a_{0,0} & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
 \tilde{A} &= \begin{pmatrix} a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ a_{0,-1} & 0 & a_{0,1} \\ a_{1,-1} & a_{1,0} & a_{1,1} \end{pmatrix}
 \end{aligned}$$

$A^0$  ve  $\tilde{A}$  ya sırasıyla şablonun merkez ve kenar bileşenleri denir.

-  $B(i, j; k, l)$  giriş operatörünün katkısı



$$\begin{aligned}
\sum_{C(k,l) \in S_r(i,j)} B(i,j;k,l)U_{kl} &= \sum_{|k-i| \leq 1} \sum_{|l-j| \leq 1} B(i-k; j-l)u_{kl} \\
&= \sum_{k=-1}^1 \sum_{l=-1}^1 b_{kl}u_{i+k, j+l}
\end{aligned} \tag{2.8}$$

$$\begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix} \otimes \begin{bmatrix} u_{-1,-1} & u_{-1,0} & u_{-1,1} \\ u_{0,-1} & u_{0,0} & u_{0,1} \\ u_{1,-1} & u_{1,0} & u_{1,1} \end{bmatrix} = B \otimes U_{ij} \tag{2.9}$$

B matrisine ileri besleme ya da giriş klonlayan şablonu denir. Benzer şekilde B şablonu ikiye ayırabilir.

$$B = B^0 + \tilde{B}$$

$$B^0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b_{0,0} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & 0 & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix}$$

Sırası ile  $B^0$  ve  $\tilde{B}$  ye ileri besleme şablonununun merkezi ve kenarı denir.

- Eşik teriminin katkısı

$$z_{ij} = z$$

Bu ifade uzayda değişmeyen bir HSA da kullanılırsa HSA ifadesi (2.10) gibi olur.

$$\dot{x}_{ij} = -x_{ij} + A \otimes Y_{ij} + B \otimes U_{ij} + z \tag{2.10}$$

(2.10) ifadesi A ve B klonlayan şablonları ayrılmış durumu ile ifade edilir

$$\dot{x}_{ij} = \underbrace{-x_{ij} + a_{00}f(x_{ij})}_{g(x_{ij})} + \underbrace{\tilde{A} \otimes Y_{ij} + B \otimes U_{ij}}_{w_{ij}(t)} + z \quad (2.11)$$

$g(x_{ij})$  : Sürücü bileşen

$w_{ij}(t)$  : Zamana bağlı, denge seviyesi.

### 2.2.3 Üç Basit HSA Sınıfı

HSA  $3 \times 3 (r=1)$  komşuluk için 19 gerçek sayı içeren klonlayan şablonlar  $\{A, B, z\}$  tarafından tanımlanır. Gerçek sayılar sayılamadıklarından dolayı, sonsuz sayıda farklı HSA şablonu vardır. Bu şablonlar basit ve matematiksel olarak 3 sınıfa ayrılır.

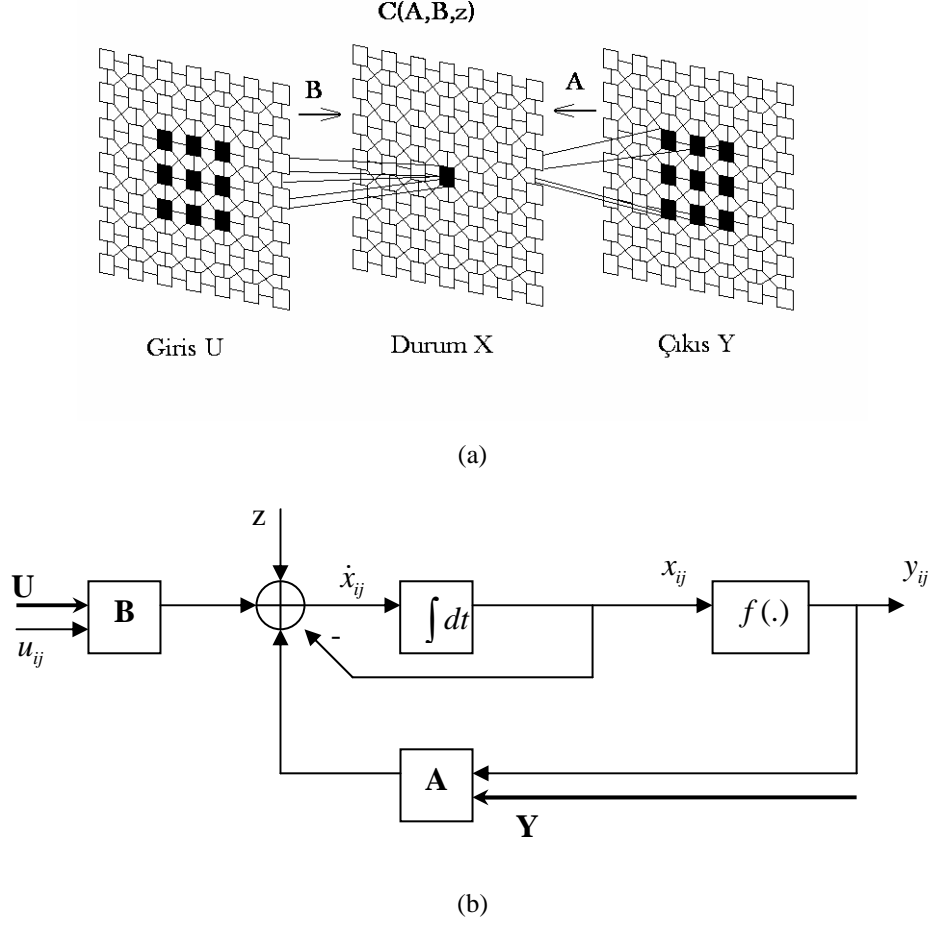
Destekleyici ve Engelleyici Sinaptik Ağırlıklar (Şekil 2.10)

Geri besleme  $a_{kl}$  sinaptik ağırlığı pozitif ise destekleyici, negatif ise engelleyici denir. Bir sinaptik ağırlık pozitif giriş için  $h_{ij}(x_{ij}, w_{ij})$  fonksiyonunu pozitif yönde etkilerse bu sinaptik ağırlık destekleyicidir.

#### 2.2.3.1 Sıfır geri besleme (İleri Besleme) sınıfı $C(0, B, z)$ (Şekil 2.11)

Eğer geri besleme şablonunun bütün elemanları sıfıra eşitse bu HSA sıfır geri besleme sınıfına  $C(0, B, z)$  dahildir.  $A \equiv 0$ . Sıfır geri besleme HSA'nın her hücresi (2.12) gösterilir;

$$\dot{x}_{ij} = -x_{ij} + B \otimes U_{ij} + z \quad (2.12)$$



Şekil 2.10  $3 \times 3 S_1(i, j)$  komşuluklu uzayda değişmeyen bir HSA. (a)  $3 \times 3$  komşuluklu HSA'nın işaret akış diyagramı. (b)  $C(i, j)$  hücresinin sistem yapısı. Koyu tonda basılmış oklar, giriş ve çıkışın çevre hücrelerinden  $u_{kl}$  ve  $y_{kl}$  gelen veri yollarıdır. İnce basılmış oklar ise sırasıyla  $z$  eşik,  $u_{ij}$  giriş,  $x_{ij}$  durum ve  $y_{ij}$  çıkıştır.

### 2.2.3.2 Sıfır giriş (Autonomous) sınıfı $C(A, 0, z)$ (Şekil 2.12)

Eğer ileri besleme şablonunun bütün elamanları sıfıra eşit ise bu HSA sıfır giriş sınıfına  $C(A, 0, z)$  aittir.  $B \equiv 0$ . Sıfır giriş sınıfı HSA'nın her bir hücresi (2.13) gibi ifade edilir.

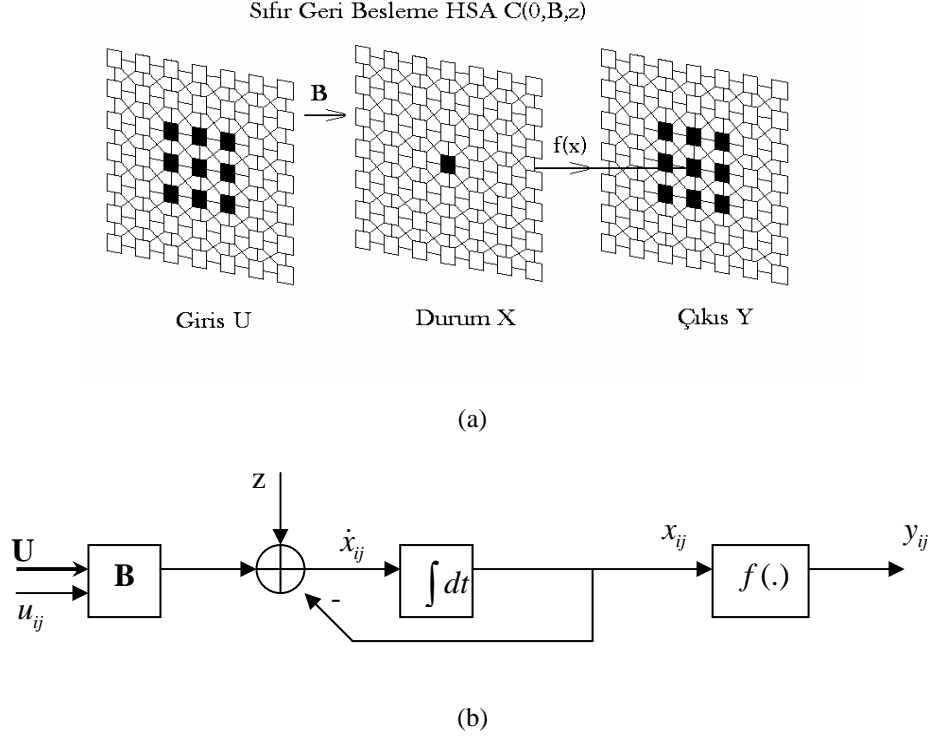
$$\dot{x}_{ij} = -x_{ij} + A \otimes Y_{ij} + z \quad (2.13)$$

### 2.2.3.3 Ayrılmamış (Sayıl-Skaler) sınıf $C(A^0, B, z)$ (Şekil 2.13)

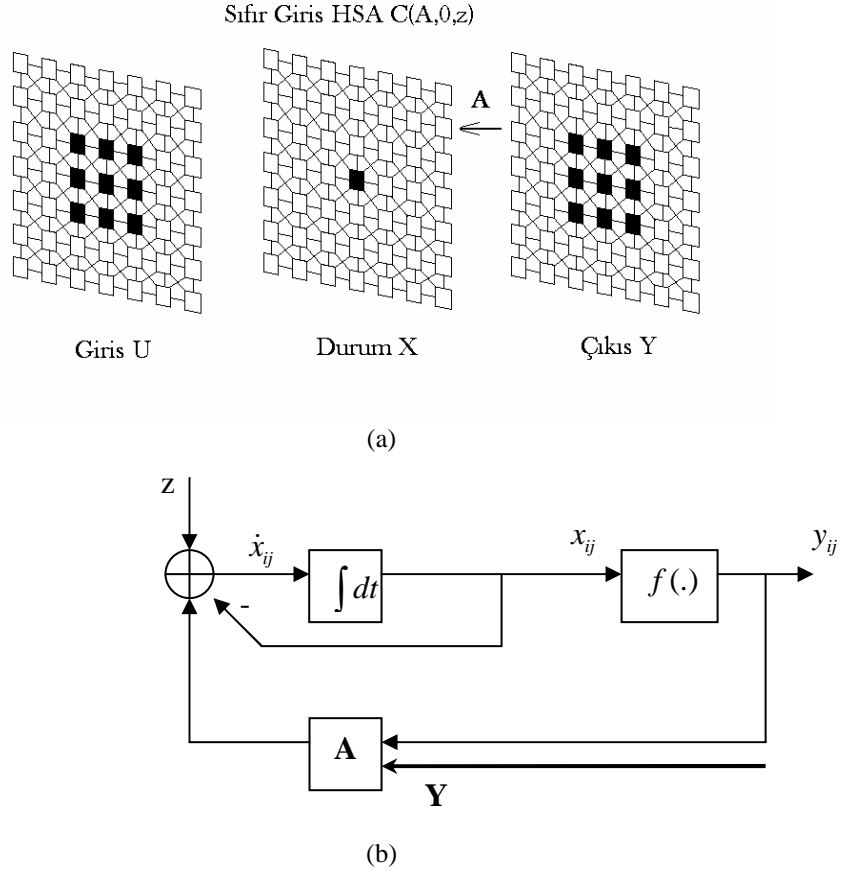
$i = j$  durumu dışındaki bütün  $a_{ij}$  değerleri sıfıra eşit ise bu HSA ayrılmamış sınıfa aittir.

$$\dot{x}_{ij} = -x_{ij} + a_{00} f(x_{ij}) + B \otimes U_{ij} + z \quad (2.14)$$

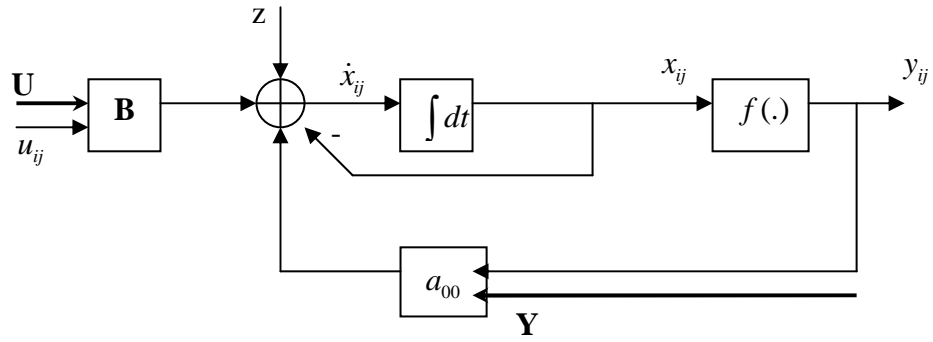
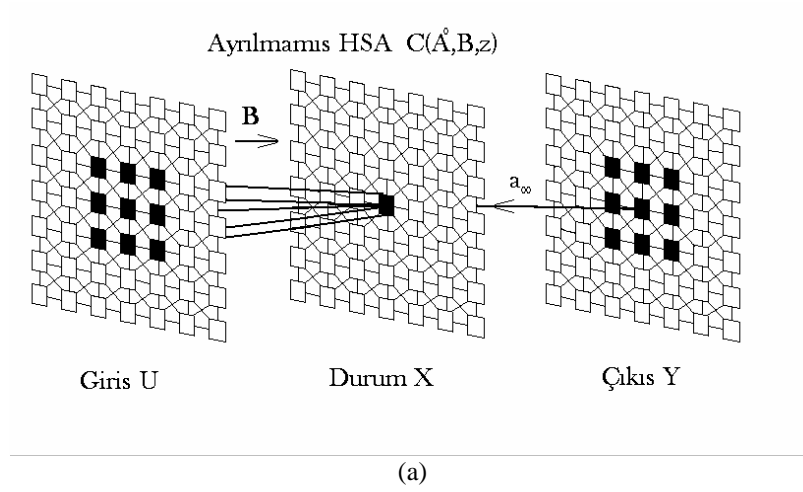
$$i \neq j \Rightarrow a_{ij} = 0$$



Şekil 2.11 Sıfır geri besleme (ileri besleme) HSA  $\in C(0, B, z)$ . (a)  $3 \times 3$  komşuluklu sıfır geri besleme HSA ın işaret akış diyagramı. (b)  $C(i, j)$  hücresinin sistem yapısı. Koyu tonda basılmış ok çevre hücrelerden gelen giriş sinyalinin gösterir. Bu durumda, kendine geri besleme ve çevre hücrelerin çıkışlarındaki kuplaj yoktur.



Şekil 2.12. Sıfır giriş (Autonomous) HSA  $\in C(A, 0, z)$ . (a)  $3 \times 3$  komşuluklu sıfır giriş HSA ın işaret akış diyagramı. (b)  $C(i, j)$  hücresinin sistem yapısı. Koyu tonda basılmış ok çevre hücrelerin çıkışlarından gelen geri beslemeyi gösterir. Bu durumda hiç giriş işareti yoktur.



Şekil 2.13 Ayrılmamış HSA  $\in C(A^0, B, z)$ . (a)  $3 \times 3$  komşuluklu ayrılmamış giriş HSA'nın işaret akış diyagramı. (b)  $C(i, j)$  hücresinin sistem yapısı. Koyu tonda basılmış oklar çevre hücrelerden gelen giriş işaretini gösterir. Bu durumda yalnızca kendi kendine geri besleme mevcuttur, çevre hücre çıkışlarında kuplaş meydana gelememektedir.

### 3. GABOR SÜZGEÇLERİ VE HSA GERÇEKLEMELERİ

#### 3.1 Giriş

Gabor süzgeci Dennis Gabor [11] tarafından geliştirilmiştir ve minimum belirsizlik ile frekans ve zaman (yada uzay) işaretlerini tanımlar. Gabor, fonksiyonu Gaussian fonksiyonunu karmaşık üstel fonksiyon ile çarparak hesaplamıştır. Daugman [12] tarafından orijinal Gabor süzgecinin iki boyutlu gösterimini yapılmıştır.

Gabor süzgeçleri bilgisayarla görme ve görüntü işleme konularındaki bir uygulama için farklı kullanımları vardır. Bunlara örnek olarak doku tanıma ve sınıflama [2], yazı tanımlama [3], doku ayrımı [4], kenar belirleme [5], görüntü sıkıştırma [6], hareket kestirimi [7], nesne tanımlama [8] ve dokuda şekil tanıma [9]. Gabor süzgecinin yön seçici özelliği kullanılarak gerçekleştirilen görsel korteks [10], bu yöntemlerin bilgisayarla görme ve sayısal görüntü işlemede kullanımını arttırmıştır. Marceljia [10] Gabor süzgecinin memelilerin görsel sistemi için kullanılabileceğini ilk öne süren kişidir. Üç boyutlu (3-D) Gabor süzgeçleri, cortical hücrelerinin hız ve yön hassasiyetinin modellenmesinde kullanılmıştır. Çünkü bir görüntüdeki hareket, zaman uzayında yer değişimi gösteren bir “zaman vektörü” olarak düşünülebilir.

Gabor süzgecinin tek eksik yanı fazla hesaplama gerektirmesidir.

#### 3.2 Gabor Süzgeçleri

Bir boyutlu (1-d) işaretlerde, Gabor süzgecinin dürtü yanıtı  $g(x)$  Gaussian işaretinin karmaşık üstel ile modüle edilmiş halidir.

$$\begin{aligned}
g(x) &= \frac{1}{\sqrt{2\pi\sigma}} e^{-(x^2/2\sigma^2)} e^{j\omega_{x0}x} \\
&= \frac{1}{\sqrt{2\pi\sigma}} e^{-(x^2/2\sigma^2)} (\cos(\omega_{x0}x) + j \sin(\omega_{x0}x))
\end{aligned} \tag{3.1}$$

$w_{x0}$  karmaşık üstelin açısal frekansı ve  $\sigma^2$  Gaussianın standart sapmasıdır. Şekil 3.1 (a) ve (b) 'de (3.1)'in gerçek ve sanal kısımları çizilmiştir. Dürtü yanıtı sürekli bir işaret için tanımlanmasına rağmen, diğer sonuçlarla karşılaştırabilmek için ayırık tamsayı değerler ile çizilmiştir.

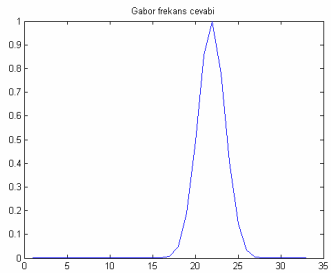
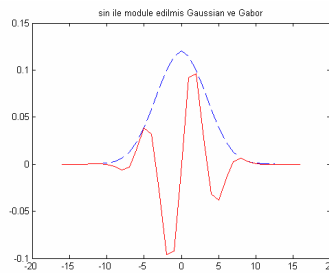
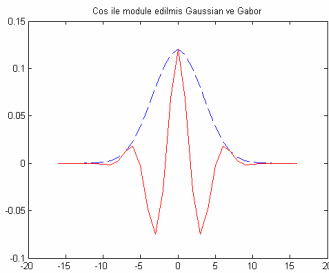
Fourier kaydırma teoremine göre Gabor süzgecinin frekans yanıtı merkezi  $(\omega_x - \omega_{x0})$  olan bir Gaussian fonksiyondur.

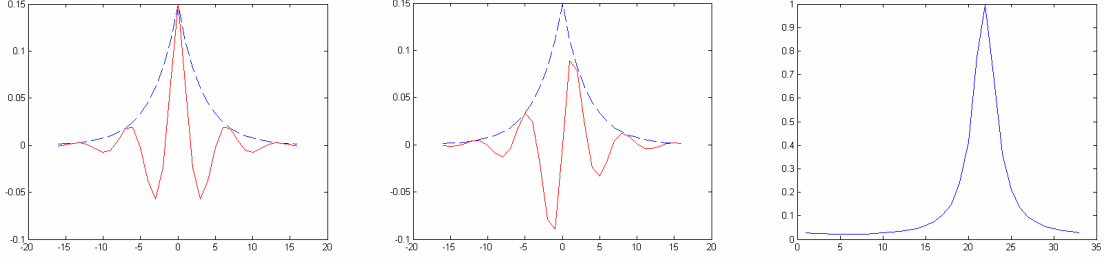
$$G(w_x) = \int_{-\infty}^{\infty} g(x) e^{-j\omega_x x} dx = e^{\left(\frac{-\sigma^2(\omega_x - \omega_{x0})^2}{2}\right)} \tag{3.2}$$

Gabor süzgeci  $w_{x0}$  ve bant genişliği içindeki frekanslara ayarlı bir bant geçiren süzgeçtir şekil (3.1 c). Gabor süzgecini n boyutlu işaretlere genellersek, konvolüsyon şablonu ve frekans yanıtı,

$$\begin{aligned}
g(\bar{x}) &= \frac{1}{\sqrt{(2\pi)^n \Delta}} e^{-(1/2)\bar{x}^T C^{-1} \bar{x}} e^{j(\bar{\omega}_{x0} \bar{x})} \quad \text{olur.} \\
G(\bar{x}) &= e^{-(1/2)(\bar{\omega}_x - \bar{\omega}_{x0})^T C (\bar{\omega}_x - \bar{\omega}_{x0})} \quad \bar{x}, \bar{\omega}_{x0} \in \mathfrak{R}^n, C \quad n \times n \text{ boyutunda ortak deęişinti} \\
&\quad \text{matrisidir. } \Delta = \det(C) \text{ dir ve pozitifdir.}
\end{aligned}$$

3-D Gabor süzgeçleri hareket analizinde kullanılır. Hareketi zaman uzamsal düzlemde yön olarak karakterize edilir.





Şekil 3.1. Gabor süzgecinin gerçek ve sanal kısmı.

### 3.2.1 İki Boyutlu (2-D) Gabor Süzgeçleri

2-D Gabor süzgeci özel bir frekansın karmaşık sinüzoidal bir yüzey dalgasıdır. Ve uzamsal bölgede 2-D Gaussianın zarfıdır, frekans bölgesinde ise 2-D Gaussian ın ötelenmiş halidir. Bu 2-D fourier dönüşümünün modülasyon özelliği kullanılarak gösterilebilir. Orijinal işaret  $f(x, y)$  karmaşık üstel  $e^{j(\omega_{x0}x + \omega_{y0}y)}$  ile çarpılır, sonucunda fourier dönüşümü yapılırsa frekans düzleminin merkezindeki işaretin  $(\omega_{x0}, \omega_{y0})$  noktasına öteleniği görülür.

$$f(x, y) \rightarrow F(\omega_x, \omega_y)$$

$$f(x, y)e^{j(\omega_{x0}x + \omega_{y0}y)} \rightarrow F(\omega_x - \omega_{x0}, \omega_y - \omega_{y0}) \quad (3.4)$$

Gabor süzgeçleri  $\mathbb{C}^2$  de karmaşık değerlidir ve bundan dolayı  $\mathbb{C}^2$  den  $\mathbb{C}$  ' ye eşleştirilerek tanımlanır.

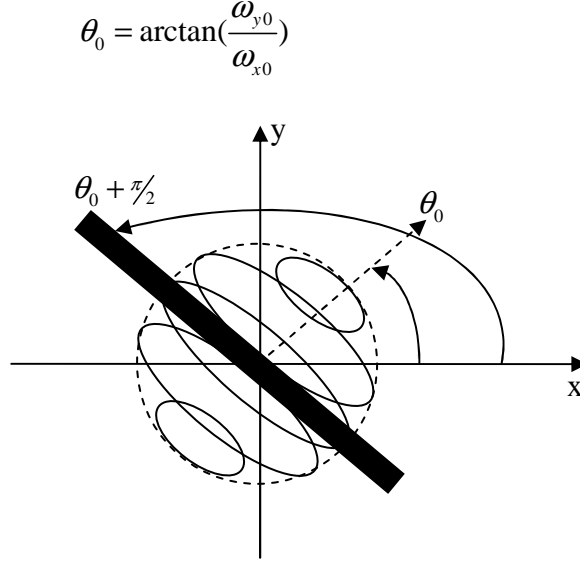
2-D boyutlu Gabor süzgeci  $(\omega_{x0}, \omega_{y0})$  frekansına ayarlı bir bant geçiren süzgeçtir. 2-D Gabor süzgecinin uzamsal darbe cevabı  $h(x, y)$  Gaussian  $g(x, y)$  fonksiyonun karmaşık üstel ile modüle edilmiş halidir.

$$h(x, y) = g(x, y)e^{j(\omega_{x0}x + \omega_{y0}y)} \quad (3.5)$$

$$g(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} \quad (3.6)$$



$(\omega_{x0}, \omega_{y0})$  karmaşık üstelin açısal frekansı ve  $\sigma^2$  Gaussianın standart sapmasıdır. 2-D Gabor süzgeci yön seçici bir süzgeçtir. Süzgecin maksimum yanıtı şekil 3.2 de gösterilen  $\theta_0 + \frac{\pi}{2}$  frekansdır.



Şekil 3.2: 2-D Gabor Süzgecinin Yön seçiciliği

Karmaşık üstelin uzamsal frekansı tercih edilen yönü, Gaussianın varyansı ise hassasiyeti belirler. 2-D Gabor süzgecinin karmaşık değerinin gerçek ve sanal kısmı yerine tek ve çift Gabor şablonları şeklinde kullanılması tercih edilir. Giriş resmi  $u(x, y)$  ile süzgeç dürtü yanıtının  $h(x, y)$  konvolüsyon toplamı  $v(x, y)$ :

$$\begin{aligned} v(x, y) &= \sum_{k,l} u(k, l) h(x-k, y-l) \\ &= \frac{1}{2\pi\sigma^2} \sum_{k,l} u(k, l) e^{-\left[\frac{(x-k)^2 + (y-l)^2}{2\sigma^2}\right]} e^{j[\omega_{x0}(x-k) + \omega_{y0}(y-l)]} \end{aligned} \quad (3.7)$$

Gaussian  $g(x, y)$  işaretinin 2-D fourier dönüşümü şu şekilde verilir:

$$G(\omega_x, \omega_y) = e^{-\frac{\sigma^2(\omega_x^2 + \omega_y^2)}{2}}$$

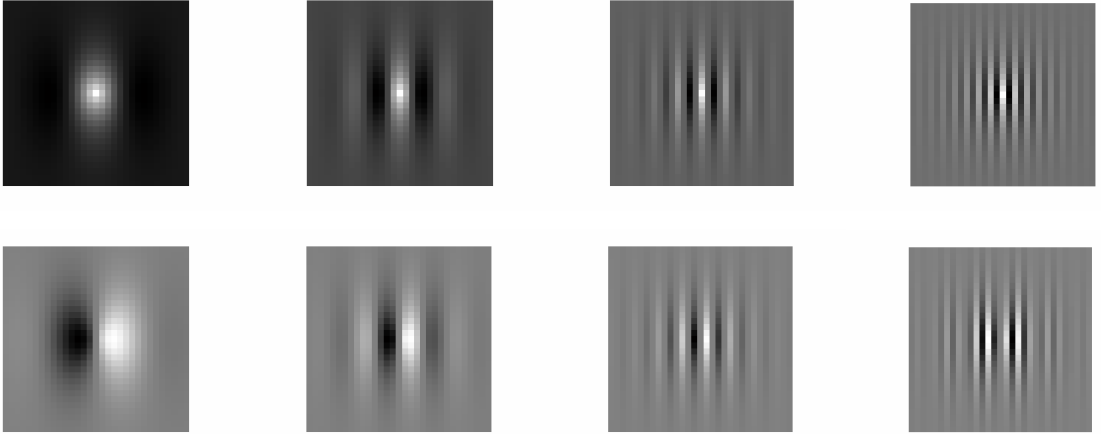
2-D fourier dönüşümünün modülasyon özelliği kullanılarak, 2-D Gaussian işaretinden merkez frekansı  $(\omega_{x0}, \omega_{y0})$  olan Gabor süzgeci elde edilirse bu Gabor süzgecinin frekans yanıtı şu şekilde gösterilir.

$$H(\omega_x, \omega_y) = e^{\frac{-\sigma^2 [(\omega_x - \omega_{x0})^2 + (\omega_y - \omega_{y0})^2]}{2}} \quad (3.8)$$

### 3.2.2 İki Boyutlu (2-D) Gabor Süzgecinin İşlemsel Ayrıntıları

Gabor süzgeci özel bir frekansa ayarlıdır ve yön seçici cevabı süzgecin uygulandığı resimde aynı uzamsal frekansı ve yönü olan elemanlardır. Gabor süzgecinin şablonunu (dürtü cevabı) göz önüne alırsak, süzgecin yönlendirilmiş barların oluşturduğu grubu aradığı düşünülebilir (şekil 3.3).

(3.5) deki  $(\omega_{x0}, \omega_{y0})$  parametreleri süzgeç cevabının daha çok uzamsal frekansını kontrol ederler.  $\sigma$  ise süzgecin bant genişliğinin ölçeklenmesi için kullanılır. Eğer karşılaştırma yapılan uzamsal frekans arttırılırsa, süzgeç tek bir bardan çok, paralel çizgilerden oluşan parçaları arar hale gelir (şekil 3.3).



Şekil 3.3: Üst sırada çift Gabor, alt sırada tek Gabor.

Tek ve çift Gabor şablonları “dördün çiftleri” olarak adlandırılır çünkü aralarında  $\pi/2$  radyan faz farkı vardır. Şablonun fazı sinüzoidal bar yapılı şablonun dikey etki alanı ile ilgilidir. Eğer etki alanı orijin noktasından geçerse faz açısı  $0^\circ$  dir. Çift Gabor şablonu özel bir yönde

simetrik elemanlardan yeniden elde edilir, tek Gabor ise asimetrik elemanlardan elde edilir. Ek olarak, tek ve çift Gabor şablonları yaklaşık olarak Hilbert çiftidir. Hilbert çiftleri denen iki dalga şeklinin eğer fourier dönüşümü alınırsa aynı genlikli fakat  $\pi/2$  faz farklı olurlar. Her ne kadar çift ve tek Gabor şablonlarının frekans yanıtları arasında  $\pi/2$  kadar faz farkı olmasına rağmen, aynı genliklere sahip değillerdir. Fakat eğer bant genişliği yeterince küçük seçilirse genlikler birbirine o kadar yaklaşır.

### 3.3 Neden HSA Gabor Tip Süzgeç?

Gabor süzgeçlerinin görüntü üzerindeki uygulamaları karmaşık yapıya sahiptir. Süzgeç katsayıları karmaşık değerlidir ve Gabor süzgecini görüntülere uygulamak için Gabor fonksiyonunun ayrık sürümünün bulunması gerekmektedir. Gabor süzgecinin hesaplama yükü ayrık düzlemdeki konvolüsyon gerekliliğini engellemektedir. Yakın bir zamanda, bu problem Shi tarafından çözülmüştür. Shi Gabor süzgecine benzeyen konvolüsyon şablonlarını bularak görüntü üzerinde uygulamıştır.

HSA ile gerçekleştirilebilen herhangi bir alçak geçiren süzgecin dürtü yanıtı karmaşık üstel bir sayı ile çarpılarak Gabor süzgeci elde edilir, bu yeni süzgeçte HSA ile gerçekleştirilebilir.

Sayısal görüntü işleme uygulamalarında kullanılan Gabor süzgeci sırasında gereken ağır hesaplama yükü analog VLSI kullanılarak azaltılabilir. HSA ile Gabor süzgecinin gerçek ve sanal kısımlarının aynı anda hesaplanabilmesi Gabor süzgecinin fazını kullanan algoritmalarda önemli bir üstünlük kazandırır.

Bundan sonraki bölümlerde HSA Gabor tip süzgeçler tanıtılacaktır.

#### 3.3.1 Bir Boyutlu HSA Gabor Tip Süzgeç

HSA hücre adı verilen nöronların oluşturduğu dizi yapısına sahip bir sinir ağı mimarisidir. Her hücre sürekli zamanda dinamik bir sistem özelliği gösterir. Süzgecin girişi olacak  $N$  piksellik tek boyutlu işaret  $u(n) \in \mathfrak{R}$  ve  $n \in \{0,1,2,\dots,N-1\}$  olarak tanımlanır.  $N$  hücreli bir HSA dizisinde  $n$ . hücrenin durumu  $v(n) \in \mathfrak{R}$  şu şekilde ifade edilir,

$$\dot{v}(n) = \sum_{k=-r}^r a_k v(n+k) + bu(n). \quad (3.10)$$

İfadedeki nokta zamanda türevi gösterir.  $A = [a_k]_{k=-r}^r$  ve  $b$  sırası ile ileri ve geri besleme klonlanan şablonların karmaşık katsayılarıdır.  $r$  komşuluk derecesidir.  $a$  geri besleme klonlanan şablonu merkez elemanı  $a_0$ 'a eşit olan  $1 \times (2r+1)$  boyutundaki bir matris ile gösterilir. Örneğin  $r = 2$  için klonlanan şablon matrisi  $A = [a_{-2} \ a_{-1} \ a_0 \ a_1 \ a_2]$  olur. Bu HSA denklemi ile [1] ve [13] kaynaklarında gösterilen HSA denklemi arasında ufak farklılıklar vardır. Karmaşık değerler yerine gerçek değerler alınmış, eşik ve lineer olmayan durumlar ele alınmamıştır. Her şeye rağmen (3.10) HSA denklemi [1] kaynağında verilen iki katmanlı HSA denkleminin özel bir hali olarak düşünülebilir. [1] referansında verilen denklemde birinci katmandaki karmaşık durum değişkenlerinin gerçek kısımları, ikinci katmanda sanal kısımları, eşik sıfır ve çıkış ise lineer olmayan çıkışın lineer bölgesi alınarak denkleme benzetilir.

(3.10) denklemindeki HSA'nın dinamikleri tamamen lineer olduğundan dolayı herhangi bir şablonun kararlılığı [14] referansında verilen teorideki gibi düşünülebilir. Eğer HSA kararlı ise zamanda durağan bir giriş için  $v(n)$  durumu  $u(n)$ 'in uzamsal olarak süzölmüş haline eşitlenir. Geçici durumun uzunluğu HSA hesaplamaları için gerekli zaman kadardır.  $-\infty < n < \infty$  arasında sonsuz hücreli bir HSA olduğu farz edilirse giriş ve çıkışın ayrık uzay fourier dönüşümü şu şekilde bulunur.

$$\begin{aligned} u(e^{j\omega_x}) &= \sum_{n=-\infty}^{\infty} u(n)e^{-j\omega_x n} \\ v(e^{j\omega_x}) &= \sum_{n=-\infty}^{\infty} v(n)e^{-j\omega_x n} \end{aligned} \quad (3.11)$$

HSA uzamsal süzgecinin frekans cevabı da

$$F(e^{j\omega_x}) = \frac{v(e^{j\omega_x})}{u(e^{j\omega_x})} = \frac{b}{-\sum_k a_k e^{jk\omega_x}} \quad (3.12)$$

olur.

Sonlu sayıda elemana sahip olan dizilerde, sınır değerleri en yakın komşu değerine eşit olarak alındığında, dizinin ayrık fourier dönüşümü de benzer sonuçlar verir. Farklı sınır koşulları içinde bu analiz hemen hemen aynı sonucu verir. Kararlı süzgeçlerde sınırlardan uzaklaştıkça sınır koşulları nedeni ile oluşan bozulmanın etkisi azalır. Klonlanan şablonun değerleri değiştirilerek alçak geçiren süzgeçten bant ve yüksek geçiren süzgeç oluşturulabilir. HSA hücre dizisi ayrık değerleri yalnızca ayrık uzay için tanımlıdır.

HSA ile gerçekleştirilebilen herhangi bir alçak geçiren süzgecin dürtü yanıtı karmaşık üstel bir sayı ile çarpılarak Gabor süzgeci elde edilir, bu yeni süzgeçte HSA ile gerçekleştirilebilir. Crouse ve Chua [15] referansında alçak geçiren süzgeç için HSA tasarım yöntemlerini sunmuştur. Bundan sonraki odaklanan nokta alçak geçiren süzgeç şablonundan Gabor süzgecinin yerini tutabilecek olan şablonun elde edilmesidir.

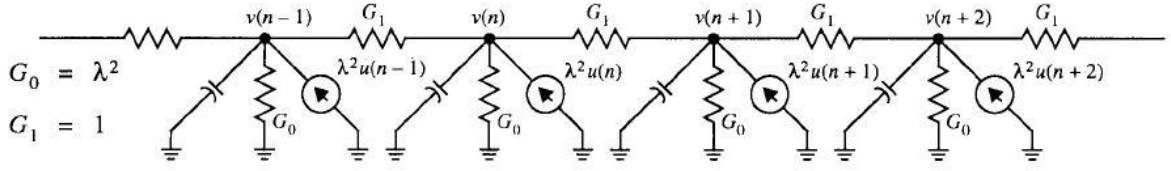
Frekans cevabı  $F(e^{j\omega_x})$ ,  $A$  ve  $B$  şablonlarından elde edilen bir HSA alçak geçiren süzgecin olduğunu düşünelim.  $F(e^{j\omega_x})$  frekans yanıtı  $\omega_{x0}$  açısal frekansına ötelendiğinde Gabor süzgecinin yerini tutabilecek bir süzgecin frekans yanıtı bulunmuş olur.

$$\begin{aligned} H(e^{j\omega_x}) &= \frac{b}{-\sum_k a_k e^{jk(\omega_x - \omega_{x0})}} \\ &= \frac{b}{-\sum_k [a_k e^{-jk\omega_{x0}}] e^{jk\omega_x}} \end{aligned} \quad (3.13)$$

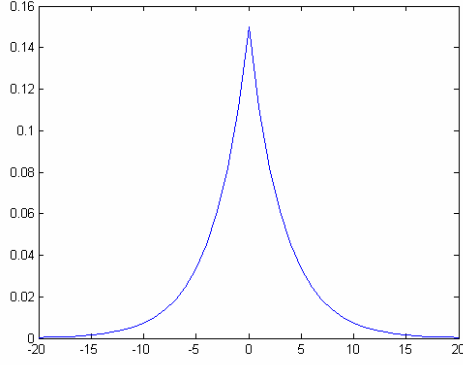
Frekans yanıtı (3.13) gibi olan bir süzgeç karmaşık değerli ileri ve geri besleme klonlanan şablonları ile oluşturulabilir.

$$A = [a_k e^{-jk\omega_{x0}}]_{k=-r}^r \quad (3.14)$$

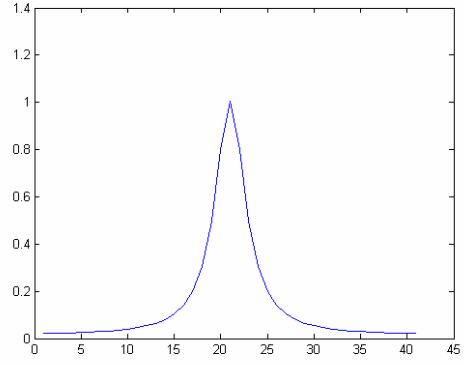
[14] kaynağındaki yöntemle göre kararlı bir alçak geçiren süzgeç şablonundan elde edilen Gabor tipi süzgecin şablonu da kararlı olur.



(a)

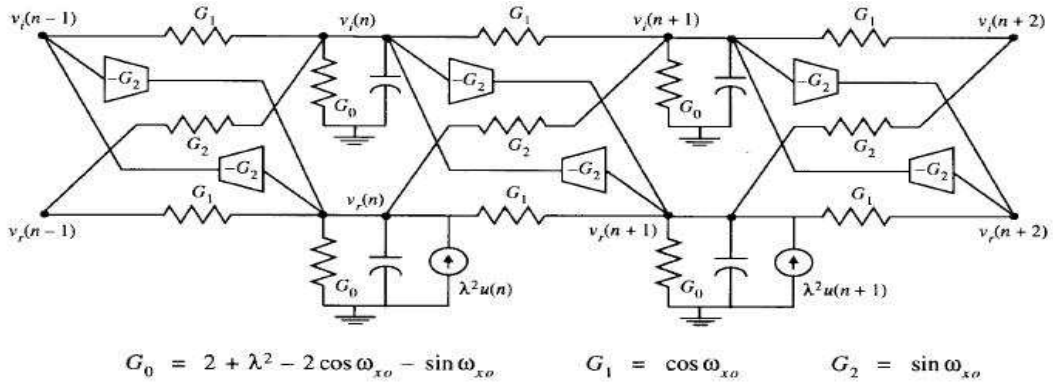


(b)



(c)

Şekil 3.4 : Tek boyutlu direnç ızgara yapılı HSA alçak geçiren süzgeç, (b) dürtü ve (c) frekans cevabı.



Şekil 3.5 : İki hücreli tek boyutlu HSA Gabor süzgecin gerçeğlenmesi

**Örnek 1:**  $A = [1 \quad -(2 + \lambda^2) \quad 1]$  ,  $b = \lambda^2$   $\lambda \geq 0$  klonlanan şablonları ile HSA da oluşturulan alçak geçiren süzgeç ele alınıyor.

$$\dot{v}(n) = -\lambda^2 v(n) + (v(n-1) - v(n)) + (v(n+1) - v(n)) + \lambda^2 u(n) \quad (3.15)$$

$v(n)$  ve  $u(n)$ 'in her ikisi de gerçek olarak kabul edilir ise, örnekteki HSA şekil (3.4 a)'da ki gibi lineer direnç ızgara yapısı olarak gerçekleştirilir. Kalıcı halde  $v(n)$ ,  $u(n)$ 'in alçak geçiren süzölmüş durumudur. Aşağıda verilen dürtü ve frekans yanıtların çizimleri şekil (3.4. b) ve (c) de gösterilmiştir.

$$f(n) = \sqrt{\frac{\lambda}{4 + \lambda}} e^{-\alpha|n|} \quad F(e^{j\omega_x}) = \frac{\lambda^2}{2 + \lambda^2 - 2 \cos \omega_x} \quad (3.16)$$

$$\alpha = a \cosh\left(2 + \frac{\lambda}{2}\right).$$

Bu sistem aşağıda dürtü ve frekans cevabı verilen sürekli uzay sisteminin ayrık bir yaklaşıklığıdır.

$$f_c(x) = \frac{\lambda}{2} e^{-\lambda|x|} \quad F_c(\omega_x) = \frac{\lambda^2}{\lambda^2 + \omega_x^2} \quad (3.17)$$

c altsimgesi sürekli uzayı vurgulamaktadır. Dürtü yanıtının uzunluğu  $1/\lambda$  ile lineer bir şekilde artar. Aynı zamanda  $\lambda$  düşer iken süzgecin yaklaşıklığı artar.

(3.14) den elde edilen  $A$  ve  $B$  klonlanan şablonları Gabor süzgecinin yerini tutan HSA şablonlarıdır.

$$A = [e^{j\omega_{x0}} \quad -(2 + \lambda^2) \quad e^{-j\omega_{x0}}] \quad b = \lambda^2 \quad (3.18)$$

Şekil (3.1 d) ve (3.1 e) de ,

$$h(n) = f(n) e^{j\omega_{x0}n} \cong \frac{\lambda}{2} e^{-\lambda|n|} e^{j\omega_{x0}n} \quad (3.19)$$

$$H(e^{j\omega_x}) = F(e^{j(\omega_x - \omega_{x0})})$$

ile verilen dürtü ve frekans yanıtları çizilmiştir.

Devre gerçekleştirilmesinde karmaşık değerli  $v(n)$  durum değişkeninin gerçek kısmı  $v_r(n)$  ile sanal kısmı  $v_i(n)$  iki kondansatörle temsil edilir. (3.18) denklemi (3.14)'de yerine konur, sanal ve gerçek kısmı ayrılır. (3.14)'deki karmaşık değerli durum değişkeninin zamanla olan değişimi ifadesinde, karmaşık durum değişkeninin yerine gerçek değerli iki durum değişkeni konarak eş bir sistem elde edilir.

$$\begin{aligned}
\begin{bmatrix} \dot{v}_r(n) \\ \dot{v}_i(n) \end{bmatrix} &= \begin{bmatrix} \cos w_{x0} & -\sin w_{x0} \\ \sin w_{x0} & \cos w_{x0} \end{bmatrix} \begin{bmatrix} v_r(n-1) \\ v_i(n-1) \end{bmatrix} - \begin{bmatrix} 2 + \lambda^2 & 0 \\ 0 & 2 + \lambda^2 \end{bmatrix} \begin{bmatrix} v_r(n) \\ v_i(n) \end{bmatrix} \\
&+ \begin{bmatrix} \cos w_{x0} & \sin w_{x0} \\ -\sin w_{x0} & \cos w_{x0} \end{bmatrix} \begin{bmatrix} v_r(n+1) \\ v_i(n+1) \end{bmatrix} + \begin{bmatrix} \lambda^2 u(n) \\ 0 \end{bmatrix}
\end{aligned} \tag{3.20}$$

Şekil 3.3'te iki hücreli bir HSA yapısı ile bu hücrelerin komşu hücreler ile olan bağlantıları verilmiştir. Düşey bağlı kondansatör çiftleri o düğüme ait olan hücrenin sanal ve gerçek kısmını gösterir. Direnç ve OTA hücreler arasındaki bağlantıları gerçekleştirirler. Ağ girişini , giriş resmindeki piksel değerleri ile orantılı akım kaynaklarında alır. Düğümlerin gerilimini  $v_r(n)$  ,  $v_i(n)$  ve kondansatör değerini de 1 olarak Kirshoff Yasasını (KCF) yazarsak, (3.20) denklemin üst kısmı  $v_r(n)$ 'in zamanla değişimini, denklemin alt tarafı ise  $v_i(n)$ 'in zamanla değişimi olduğunu görülür. Bu HSA yapısına daha fazla hücre ve bağlantı ekleyerek daha büyük bir yapı oluşturulabilir. Kalıcı durumda, alttaki kondansatörler üzerindeki gerilim ,  $h(n)$  ile giriş işaretinin (akımının) uzamsal konvolüsyonudur. Üstteki kondansatörler üzerindeki gerilim ise  $h(n)$  ile giriş işaretinin (akımının) uzamsal konvolüsyonuna eşittir. Çapraz bağlı ota ve direnç düzeneği, sadece direnç yada yalnız ota tabanlı yapılardan parametre dönüşümleri bakımından daha iyidir.

HSA'nın çalışma mantığının daha iyi bir şekilde kavranabilmesi için devre yapısı ile verilmiştir. Giriş işaretinin n. elemanın birim dürtü olduğunu farz edelir. Devrede bu duruma göre  $\lambda^2 u(n)$  akım kaynağının değeri  $\lambda^2$  ve diğer akım kaynaklarının değerleri ise sıfır olur. Eğer kazanç ve iletkenlik değerleri  $\lambda = 0,3$  ve  $\omega_{x0} = 0,93$  değerlerine göre seçilirse, kondansatörler üzerindeki kalıcı durum gerilimleri n. hücrede oluşan merkez zirve yi uzamsal düzlemde takip ederler. Devreye giren akımın  $\lambda^2 u(n)$  olduğu bilindiği için bu durumun nasıl oluştuğu düşünülebilir.  $G_0$  iletkeni üzerinden akımın bir kısmı  $v_r(n)$  kaynaklıdır. Bu voltaj değeri arttığı zaman  $G_1 = \cos \omega_{x0}$  değerli iki direnç üzerinden  $v_r(n-1)$  ve  $v_r(n+1)$  gerilimlerini  $v_r(n)$ 'ye doğru çekerek bulur etkisine neden olur. Değeri  $G_2 = \sin \omega_{x0}$  olan çapraz direnç üzerinden akan akım  $v_i(n+1)$  gerilimini pozitif çeker. Aynı anda ,  $v_r(n)$  ota ile  $v_i(n-1)$ 'den akım çekerek  $v_i(n-1)$  düğümünün gerilimini negatifleştirir. Daha büyük  $G_2$  değerleri  $v_i(n-1)$  değerini negatife,  $v_i(n+1)$  değerini ise pozitif çeker. Diğer taraftan daha büyük  $G_1$  değeri bulurluğu artırır.



### 3.3.2 İki Boyutlu (2-D) HSA Gabor Tip Süzgeçler

Gabor tip süzgeç alçak geçiren bir süzgecin karmaşık bir üstel ile modüle edilmesi sonucunda oluşan karmaşık değerli dürtü yanıtına sahiptir. Gabor süzgeci karmaşık üstelin salınım yaptığı doğrultuyu seçtiğinden ötürü yön seçicidir. Gabor tip süzme için gereken HSA şablonlarının bu teorik çerçeve içinde tasarımı Shi tarafından yapılmıştır [16]. Shi klonlayan şablonların katsayılarının karmaşık olacağını göstermiştir. Dolayısıyla standart HSA modelinden [13] Gabor tip süzgeç elde edilebilir.

$M \times N$  boyutundaki bir resmin Gabor tip uzamsal süzgeç mimarisi,  $u(m, n) \in \mathbb{C}$ ,  $m \in \{0, 1, \dots, M-1\}$  ve  $n \in \{0, 1, \dots, N-1\}$ ,  $M \times N$  boyutundaki hücreler dizisini içerir. Resimdeki her piksel bir hücreye denk gelir.  $(m, n)$ . hücredeki  $v(m, n)$  durumu şu şekilde gösterilir.

$$\hat{v}(m, n) = \sum_{k, l=-r}^r a_{k, l} v(m+k, n+l) + bu(m, n) \quad (3.21)$$

Buradaki  $(\hat{\cdot})$  zamanda türevi göstermektedir.  $A = [a_{k, l}]_{k, l=-r}^r$  ve  $b$  sırası ile geri besleme ve ileri besleme klonlayan şablonlarıdır ve  $r$  komşuluğu göstermektedir. Geri besleme klonlayan şablonu merkez elemanı  $a_{0,0}$  olan  $(2r+1) \times (2r+1)$  boyutundaki bir matristir.  $r=1$  için klonlayan şablonlar şu şekli alırlar.

$$A = \begin{bmatrix} a_{-1,1} & a_{0,1} & a_{1,1} \\ a_{-1,0} & a_{0,0} & a_{1,0} \\ a_{-1,-1} & a_{0,-1} & a_{1,-1} \end{bmatrix} ; B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.22)$$

(3.21) HSA denklemi ile [1] ve [13] kaynaklarında gösterilen HSA denklemi arasında ufak farklılıklar vardır. Karmaşık değerler yerine gerçek değerler alınmış, eşik ve lineer olmayan durumlar ele alınmamıştır. Her şeye rağmen (3.21) HSA denklemi [13] kaynağında sunulan iki katmanlı HSA denkleminin özel bir hali olarak düşünülebilir. [13] referansında verilen denklemde birinci katmandaki karmaşık durum değişkenlerinin gerçek kısımları, ikinci katmanda sanal kısımları, eşik sıfır ve çıkış ise lineer olmayan çıkışın lineer bölgesi alınarak (3.21) denklemine benzetilir. Klonlayan şablonların katsayı değerleri değiştirilerek farklı

süzgeçler elde edilebilir, örnek olarak alçak geçiren, yüksek geçiren, bant geçiren gibi. Alçak ve yüksek geçiren süzgeçler için klonlayan şablon katsayıları gerçek olur.

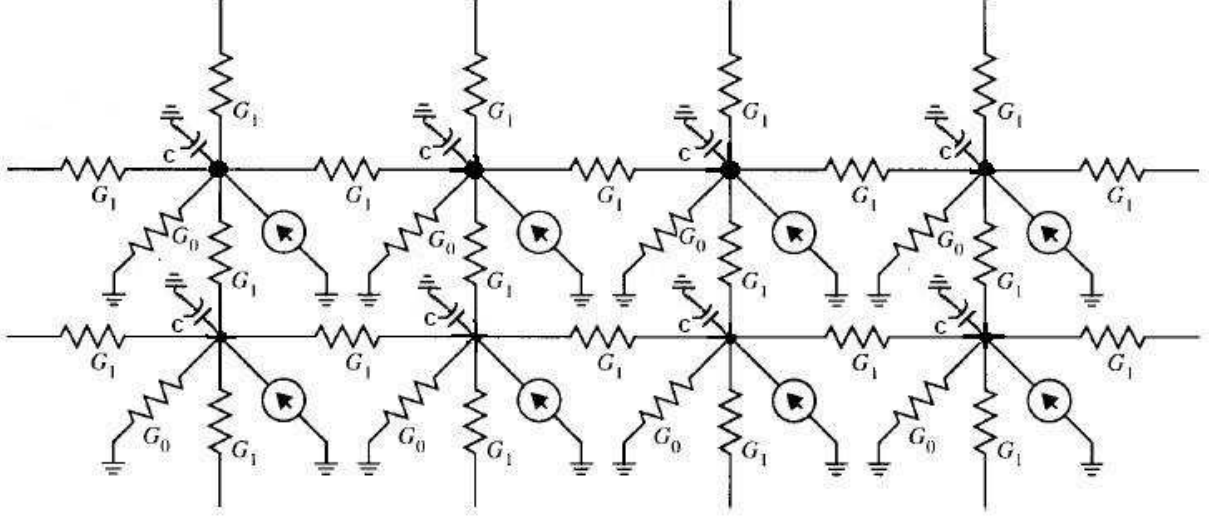
HSA ile gerçekleştirilebilen herhangi bir alçak geçiren süzgecin dürtü yanıtı karmaşık üstel bir sayı ile çarpılarak Gabor süzgeci elde edilir, bu yeni süzgeçte HSA ile gerçekleştirilebilir. Crouse ve Chua [17] referansında alçak geçiren süzgeç için HSA tasarım yöntemlerini sunmuştur.

Giriş  $u(m,n)$  ve çıkış  $v(m,n)$  gerçek değerli, şekil 3.5'te verilen girişi  $u(m,n)$  olan bir lineer direnç ızgara yapısı göz önüne alınsın. lineer ızgara yapısını her düğüme bir kondansatör  $C$  ve toprak düzeneği koyarak HSA ya dönüştürülür. kondansatörlerin kalıcı durumda direnç ızgara yapısına bir etkisi yoktur, sadece devrenin zamansal dinamiklerine etki eder. Pratik hayattaki direnç ızgara yapısı gerçeklemelerinde zaten parazitik kondansatörler vardır.

Şekilde  $u(m,n)$  akımı girişi ve  $v(m,n)$  gerilimi ise süzgecin çıkışını temsil etmektedir. Dirençlerin isimleri iletkenlikleri gösterir ve değerleri  $G_0 = \lambda^2, G_1 = 1$  olarak verilir ( $\lambda$  gerçek bir sabittir). Her düğüm ile toprak arasına yerleştirilen kondansatörler yerine birim kondansatörler ( $C=1$ ) konulmuştur. Bu direnç ızgara yapısı ile  $r=1$  bağlantı yarıçapı ile HSA modellenmesi gösterilecektir.

Şekil 3.6 teki  $(m,n)$  düğümü için Kirchoff Yasası (KCL) yazılırsa

$$\begin{aligned}
 & [v(m-1,n) - v(m,n)]G_1 + [v(m,n-1) - v(m,n)]G_1 + \\
 & [v(m+1,n) - v(m,n)]G_1 + [v(m,n+1) - v(m,n)]G_1 - \\
 & v(m,n)G_0 - C \frac{dv(m,n)}{dt} + \lambda^2 u(m,n) = 0
 \end{aligned} \tag{3.23}$$



Şekil 3.6 : HSA alçak geçiren süzgecin lineer direnç ızgara yapılı devre gerçekleştirilmesi

(3.23) daki değerler yerlerine konu ve yeniden düzenlenirse,

$$\dot{v}(m,n) = \frac{dv(m,n)}{dt} = -(4 + \lambda^2)v(m,n) + v(m-1,n) + v(m,n-1) + v(m+1,n) + v(m,n+1) + \lambda^2 u(m,n) \quad (3.24)$$

(3.24) ile (3.21) ve (3.22) karşılaştırılırsa, alçak geçiren HSA süzgeci için klonlayan şablonlar şöyle bulunur:

$$A = \begin{bmatrix} a_{-1,1} & a_{0,1} & a_{1,1} \\ a_{-1,0} & a_{0,0} & a_{1,0} \\ a_{-1,-1} & a_{0,-1} & a_{1,-1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -(4 + \lambda^2) & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \lambda^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.25)$$

HSA boyutu sonsuz alırsak ve bu HSA (3.21) ya 2-D ayrık Fourier dönüşümünü uygularsak, giriş  $u(m,n)$  ve kalıcı durum  $v(m,n)$  tanımlanabilir. Girişin  $u(m,n)$  ve kalıcı durumun  $v(m,n)$  2-D ayrık Fourier dönüşümü:

$$\begin{aligned}
U(e^{j\omega_x}, e^{j\omega_y}) &= \sum_{m,n=-\infty}^{\infty} u(m,n) e^{-j(\omega_x m + \omega_y n)} \\
V(e^{j\omega_x}, e^{j\omega_y}) &= \sum_{m,n=-\infty}^{\infty} v(m,n) e^{-j(\omega_x m + \omega_y n)}
\end{aligned} \tag{3.26}$$

$0 \leq \omega_x, \omega_y \leq 2\pi$  arasındaki uzamsal frekanslarda tanımlanır. Ayrık uzay Fourier dönüşümünün öteleme özelliği kullanılarak aşağıdaki ifade yazılabilir.

$$\begin{aligned}
f(m,n) &\rightarrow F(e^{j\omega_x}, e^{j\omega_y}) \\
f(m \mp k, n \mp l) &\rightarrow e^{\mp j(\omega_x k + \omega_y l)} F(e^{j\omega_x}, e^{j\omega_y})
\end{aligned} \tag{3.27}$$

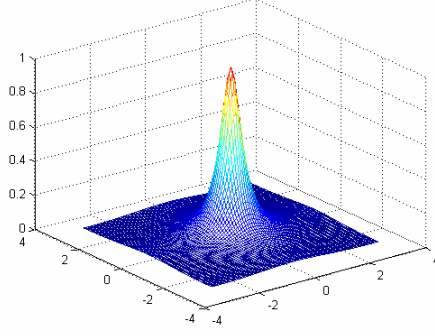
Kalıcı durumda giriş  $u(m,n)$  ve durumdan  $v(m,n)$  elde edilen HSA alçak geçiren süzgecin transfer fonksiyonu (frekans yanıtı) şöyledir:

$$H_{LP}(e^{j\omega_x}, e^{j\omega_y}) = \frac{V(e^{j\omega_x}, e^{j\omega_y})}{U(e^{j\omega_x}, e^{j\omega_y})} = \frac{b}{-\sum_{k,l=-r}^r a_{k,l} e^{j(\omega_x k + \omega_y l)}} \tag{3.28}$$

(3.25) de verilen klonlayan şablon katsayıları (3.28) da yerlerine konursa sonuç

$$\begin{aligned}
H_{LP}(e^{j\omega_x}, e^{j\omega_y}) &= \frac{\lambda^2}{4 + \lambda^2 - e^{-j\omega_x} - e^{-j\omega_y} - e^{j\omega_x} - e^{j\omega_y}} \\
&= \frac{\lambda^2}{4 + \lambda^2 - 2 \cos \omega_x - 2 \cos \omega_y}
\end{aligned} \tag{3.29}$$

HSA alçak geçiren süzgecin  $\lambda = 0.4$  için frekans yanıtının genliği şekil 3.7 da gösterilmiştir.



Şekil 3.7: HSA alçak geçiren süzgecin  $\lambda = 0.4$  için frekans yanıtı

Frekans yanıtı  $H_{LP}(e^{j\omega_x}, e^{j\omega_y})$  olan HSA alçak geçiren süzgeci  $(\omega_{x0}, \omega_{y0})$  frekansına ötelenirse, frekans yanıtı (3.30) gibi olan HSA Gabor tip süzgeç elde edilir.

$$\begin{aligned}
 H_{BP}(e^{j\omega_x}, e^{j\omega_y}) &= H_{LP}(e^{j(\omega_x - \omega_{x0})}, e^{j(\omega_y - \omega_{y0})}) \\
 &= \frac{b}{-\sum_{k,l=-r}^r a_{k,l} e^{j(k(\omega_x - \omega_{x0}) + l(\omega_y - \omega_{y0}))}} \\
 &= \frac{b}{-\sum_{k,l=-r}^r [a_{k,l} e^{-j(k\omega_{x0} + l\omega_{y0})}] e^{j(k\omega_x + l\omega_y)}}
 \end{aligned} \tag{3.30}$$

2-D HSA alçak geçiren süzgecinde kullanılan geri besleme klonlayan şablonlarından  $(\omega_{x0}, \omega_{y0})$  frekansına ayarlı 2-D HSA Gabor tip süzgecinin geri besleme klonlayan şablonu, gerçek değerler yerine karmaşık değerler konarak elde edilir.

$$A = [a_{k,l}]_{k,l=-r}^r \text{ den,} \tag{3.31}$$

$$A = [a_{k,l} e^{-j(k\omega_{x0} + l\omega_{y0})}]_{k,l=-r}^r \text{ elde edilir.} \tag{3.32}$$

(3.25) denklemi (3.32) denkleminde yerine konulduğunda, 2-D Gabor tip süzgecinin geri besleme klonlayan şablonu şu şekilde bulunur:

$$A = \begin{bmatrix} 0 & e^{-j\omega_{y0}} & 0 \\ e^{j\omega_{x0}} & -(4 + \lambda^2) & e^{-j\omega_{x0}} \\ 0 & e^{j\omega_{y0}} & 0 \end{bmatrix} \tag{3.33}$$

Eğer orijinal alçak geçiren süzgecin şablonları kararlı ise elde edilen Gabor tip süzgecin şablonları da kararlı olur.

$(\omega_{x0}, \omega_{y0})$  merkez frekansına ayarlı bu süzgecin uzamsal frekans genliği  $\Omega_0 = \sqrt{\omega_{x0}^2 + \omega_{y0}^2}$  ve yönü ise  $\theta_0 = \arctan(\omega_{y0}/\omega_{x0})$  dır.

### 3.3.3 İki Boyutlu (2-D) HSA Gabor Tip Süzgecin Dürtü Yanıtı, Frekans Yanıtı ve Bant Genişliği

İlk önce HSA alçak geçiren süzgecin dürtü yanıtı  $h_{LP}(m, n)$  elde edilir. Sonra Fourier dönüşümünün modülasyon özelliği kullanılarak HSA Gabor tip süzgecin dürtü yanıtı (3.34) gibi bulunur.

$$h_{BP}(m, n) = h_{LP}(m, n)e^{j(\omega_{x0}m + \omega_{y0}n)} \quad (3.34)$$

$\cos(\omega_x)$  ve  $\cos(\omega_y)$  nin Maclaurin serisine açarak ilk iki terimi alınır:

$$\begin{aligned} \cos(\omega_x) &\approx 1 - \frac{\omega_x^2}{2!} \\ \cos(\omega_y) &\approx 1 - \frac{\omega_y^2}{2!} \end{aligned} \quad (3.35)$$

(3.35) denklemini (3.29) denkleminde yerine konularak HSA alçak geçiren süzgecin yaklaşık bir frekans yanıtı bulunur (3.36):

$$H_{LP}(e^{j\omega_x}, e^{j\omega_y}) \approx \frac{\lambda^2}{\lambda^2 + \omega_x^2 + \omega_y^2} \quad (3.36)$$

(3.35) ile yapılan bu yaklaşıklığın tanımlı olduğu bölge  $-\pi/2 \leq \omega_x, \omega_y \leq \pi/2$  arasındadır.

Bundan dolayı (3.36) denklemini ile verilen bu frekans yanıtı yalnızca küçük bant genişliklerinde geçerlidir.

$h_{LP}(m, n)$  yaklaşıklığı için, dairesel simetrik sürekli uzay  $H_{LP}(\omega_x, \omega_y)$  süzgecinin transfer fonksiyonuna yaklaşık olarak eşit olan  $H_{LP}(e^{j\omega_x}, e^{j\omega_y})$  göz önüne alınacaktır.

$$H_{LP}(\omega_x, \omega_y) = \frac{\lambda^2}{\lambda^2 + \omega_x^2 + \omega_y^2} \quad (3.37)$$

Bu durumda sürekli uzay süzgecinin dürtü yanıtı  $h_{LP}(x, y)$  aşağıda açıklanan Hankel dönüşümü ile bulunur.

Dairesel simetrik bir sinyal her zaman kutup koordinat sisteminde yazılabilir.

$$\begin{aligned} f(x, y) &= f(r, \theta) = f(r) \\ r^2 &= x^2 + y^2 \\ x &= r \cos(\theta) \\ y &= r \sin(\theta) \end{aligned} \quad (3.38)$$

Dairesel simetrik olan  $f(r)$  işaretinin Fourier dönüşümü  $F(\omega_x, \omega_y)$  bir Hankel dönüşümü  $F(\Omega)$  olarak yazılabilir.

$$F(\omega_x, \omega_y) = 2\pi F(\Omega) = 2\pi \int_0^{\infty} f(r) J_0(\Omega r) r dr \quad (3.39)$$

$$\begin{aligned} \Omega^2 &= \omega_x^2 + \omega_y^2 \\ \omega_x &= \Omega \cos(\theta) \\ \omega_y &= \Omega \sin(\theta) \end{aligned} \quad (3.40)$$

$J_0(\cdot)$  Bessel fonksiyonudur. Henkel ters dönüşümü [18] referansında verildiği gibi:

$$f(r) = \int_0^{\infty} F(\Omega) J_0(\Omega r) \Omega d\Omega \quad (3.41)$$

$H_{LP}(\omega_x, \omega_y)$  nin ters 2-D Fourier dönüşümünü (Sürekli uzay süzgecinin dürtü cevabı  $h_{LP}(x, y)$ ),  $H_{LP}(\omega_x, \omega_y)$  yi kutup koordinat sisteminde yazarak ve daha sonra ters Hankel dönüşümü alarak bulabiliriz.

$$\begin{aligned} H_{LP}(\Omega) &= \frac{1}{2\pi} H_{LP}(\omega_x, \omega_y) \\ &= \frac{1}{2\pi} \frac{\lambda^2}{\lambda^2 + \Omega^2} \end{aligned} \quad (3.42)$$

$H_{LP}(\Omega)$  (3.41) te yerine konur ve [18] referansında verilen Hankel dönüşüm tablosu kullanılırsa sonuç aşağıdaki sonuç çıkar.

$$\begin{aligned} h_{LP}(r) &= \int_0^{\infty} H_{LP}(\Omega) J_0(\Omega r) \Omega d\Omega \\ &= \frac{\lambda^2}{2\pi} \int_0^{\infty} \frac{1}{\lambda^2 + \Omega^2} J_0(\Omega r) \Omega d\Omega \\ &= \frac{\lambda^2}{2\pi} K_0(\lambda r) \end{aligned} \quad (3.43)$$

$K_0(\cdot)$  düzenlenmiş sıfır sıralı ikinci tür bir Bessel fonksiyonudur. Kartezyen koordinatlarda yazılmış (3.43) denkleminde, sürekli uzay süzgecinin dürtü yanıtı bulunur.

$$h_{LP}(x, y) = \frac{\lambda^2}{2\pi} K_0(\lambda \sqrt{x^2 + y^2}) \quad (3.44)$$

HSA dizisinin dürtü yanıtı (3.44) denkleminde verilen sürekli uzay dürtü yanıtının ayrık bir yaklaşımıdır. O halde, HSA alçak geçiren süzgecin ayrık dürtü yanıtı aşağıdaki gibi verilebilir.

$$h_{LP}(m, n) \approx \frac{\lambda^2}{2\pi} K_0(\lambda \sqrt{m^2 + n^2}) \quad (3.45)$$

(3.45) de verilen fonksiyon orijin noktasında sonsuz olduğundan dolayı, ayrık dürtü yanıtı  $(m, n) \neq (0, 0)$  için yaklaşık olarak tanımlanır. Bu yaklaşımda orijin noktasındaki  $h_{LP}(m, n)$



ayrık dürtü yanıtının değeri, (3.24) de verilen alçak geçiren HSA süzgecinin durum denkleminin orijin noktasındaki  $v(m, n)$  nin kalıcı durum değeridir.

$$\frac{dv(0,0)}{dt} = -(4 + \lambda^2)v(0,0) + v(-1,0) + v(0,-1) + v(1,0) + v(0,1) + \lambda^2 u(0,0) \quad (3.46)$$

(3.45) de verilen  $h_{LP}(m, n)$  ayrık dürtü yanıtı (3.24) de verilen eşitliğin  $u(m, n) = 1$  için çözümüdür. Bundan dolayı (3.46) eşitliğinde  $v(.,.)$  yerine  $h_{LP}(.,.)$  konulursa ve  $\frac{dv(0,0)}{dt} = 0$  (kalıcı durum) alınırsa sonuç,

$$h_{LP}(-1,0) = h_{LP}(0,-1) = h_{LP}(1,0) = h_{LP}(0,1) = \frac{\lambda^2}{2\pi} K_0(\lambda) \quad (3.47)$$

olur. (3.45) göz önüne alınırsa aşağıdaki eşitlikler çıkarılabilir.

$$(4 + \lambda^2)h_{LP}(0,0) = h_{LP}(-1,0) + h_{LP}(0,-1) + h_{LP}(1,0) + h_{LP}(0,1) + \lambda^2 \quad (3.48)$$

(3.48) daki değerler (3.47) eşitliğinde yerlerine konursa şu sonuç elde edilir.

$$h_{LP}(0,0) = \frac{\lambda^2}{4 + \lambda^2} \left[ 1 + \frac{2}{\pi} K_0(\lambda) \right] \quad (3.49)$$

Bu sonuca göre, HSA alçak geçiren süzgecin ayrık dürtü yanıtı  $h_{LP}(m, n)$  şu şekilde verilebilir.

$$h_{LP}(m, n) = \begin{cases} \frac{\lambda^2}{2\pi} K_0(\lambda \sqrt{m^2 + n^2}) & (m, n) \neq (0, 0) \\ \frac{\lambda^2}{4 + \lambda^2} \left[ 1 + \frac{2}{\pi} K_0(\lambda) \right] & (m, n) = (0, 0) \end{cases} \quad (3.50)$$

(3.50) eşitliği (3.34) da yerine konursa 2-D HSA Gabor tip süzgecin ayrık dürtü yanıtı bulunur.

$$h_{BP}(m, n) = \begin{cases} \frac{\lambda^2}{2\pi} K_0(\lambda\sqrt{m^2 + n^2}) e^{j(\omega_x m + \omega_y n)} & (m, n) \neq (0, 0) \\ \frac{\lambda^2}{4 + \lambda^2} \left[ 1 + \frac{2}{\pi} K_0(\lambda) \right] e^{j(\omega_x m + \omega_y n)} & (m, n) = (0, 0) \end{cases} \quad (3.51)$$

Şimdi ise 2-D ayrık Fourier dönüşümü yerine, 2-D Z dönüşümü yaparak 2-D HSA Gabor tip süzgecin frekans yanıtı bulunabilir. Eğer süzgeç kararlı ise süzgeç osilasyon yapmayacaktır ve (3.21) eşitliği kararlı bir denge noktasına varacaktır.

$$\sum_{k, l=-1}^1 a_{k, l} v(m+k, n+l) + bu(m, n) = 0 \quad (3.52)$$

Bu şart altında aşağıdaki şekilde bir eşitlik yazılabilir.

$$-bu(0, 0) = a_{-1, 0} v(m-1, n) + a_{0, 1} v(m, n+1) + a_{0, 0} v(m, n) + a_{1, 0} v(m+1, n) + a_{0, -1} v(m, n-1) \quad (3.53)$$

(3.33) deki geri besleme klonlayan şablonu ve (3.25) deki iler besleme klonlayan şablonu kullanılırsa, şu elde edilir.

$$-\lambda^2 u(0, 0) = e^{j\omega_x} v(m-1, n) + e^{-j\omega_y} v(m, n+1) - (4 + \lambda^2) a_{0, 0} v(m, n) + e^{-j\omega_x} a_{1, 0} v(m+1, n) + e^{j\omega_y} a_{0, -1} v(m, n-1) \quad (3.54)$$

(3.54) eşitliğinin iki tarafının 2-D Z dönüşümü sonucu,

$$V(z_m, z_n) \left[ (4 + \lambda^2) - e^{j\omega_x} z_m^{-1} - e^{-j\omega_y} z_n - e^{-j\omega_x} z_m - e^{j\omega_y} z_n^{-1} \right] = \lambda^2 U(z_m, z_n) \quad (3.55)$$

(3.55) eşitliğine  $z_m = e^{j\omega_x}$  ve  $z_n = e^{j\omega_y}$  değerleri verilirse, 2-D HSA Gabor tip süzgecin frekans yanıtı bulunur.

$$H_{BP}(e^{j\omega_x}, e^{j\omega_y}) = \frac{V(e^{j\omega_x}, e^{j\omega_y})}{U(e^{j\omega_x}, e^{j\omega_y})} = \frac{\lambda^2}{4 + \lambda^2 - 2 \cos(\omega_x - \omega_{x0}) - 2 \cos(\omega_y - \omega_{y0})} \quad (3.56)$$

(3.56) de verilen frekans yanıtı (3.30) de verilen eşitlikte klonlayan şablonların katsayıları konarak ta elde edilebilir.

$\omega_x, \omega_{x0}$  ve  $\omega_y, \omega_{y0}$  değerleri birbirine yakın olması durumunda  $\cos(\omega_x - \omega_{x0})$  ve  $\cos(\omega_y - \omega_{y0})$  in Taylor serisine açarak ilk iki terimi alınır ve bu alınan terimler (3.56) eşitliğinde yerlerine konursa frekans yanıtının yaklaşık bir değeri bulunur.

$$H_{BP}(e^{j\omega_x}, e^{j\omega_y}) \approx \frac{\lambda^2}{\lambda^2 + (\omega_x - \omega_{x0})^2 + (\omega_y - \omega_{y0})^2} \quad (3.57)$$

Gabor tip süzgecin geçirme bandı uzamsal frekans düzleminde frekans yanıtının maksimum cevabının genlik değerinin yarısı ile maksimum değer arasındaki bölgedir (-6dB çevrit). Eğer  $\omega_{xc}$  ve  $\omega_{yc}$  kesim frekansları olarak tanımlarsak, 6dB çevritin geometrik yerini (3.57) den bulunabilir.

$$\left| H_{BP}(e^{j\omega_x}, e^{j\omega_y}) \right| = \frac{\lambda^2}{\lambda^2 + (\omega_x - \omega_{x0})^2 + (\omega_y - \omega_{y0})^2} = \frac{1}{2} \quad (3.58)$$

$$(\omega_x - \omega_{x0})^2 + (\omega_y - \omega_{y0})^2 = \lambda^2$$

Bu denklem merkezi  $(\omega_{x0}, \omega_{y0})$  olan ve yarıçapı  $\lambda$  olan bir çember tanımlar. Böylece ,  $(\omega_{x0}, \omega_{y0})$  ve  $(\omega_{xc}, \omega_{yc})$  yakın değerleri için, 6dB çevrit bir dairedir ve 2-D HSA Gabor tip süzgeç ise dairesel bir frekans yanıtına sahiptir.  $\lambda$  değeri süzgecin yarım bant genişliğine eşittir. Oktav olarak Gabor süzgecin bant genişliği, süzgecin geçirme bandının alt ve üst kesim frekans oranının iki tabanlı logaritması olarak tanımlanır. Örneğin, 1 oktavlık bant genişliğine sahip bir Gabor süzgecin yüksek kesim frekansı alçak kesim frekansından iki kat büyüktür.

$B_x$  ve  $B_y$  bant genişliklerini ,  $Q_x$  ve  $Q_y$  ise x ve y yönlerindeki kalite faktörünü temsil etsin. (2.58) eşitliğinden kolay bir şekilde  $B_x = B_y = 2\lambda$  olduğu görülür. Böylece, x ve y yönlerindeki kalite faktörü aşağıdaki gibi olur.

$$Q_x = \frac{\omega_{x0}}{B_x} = \frac{\omega_{x0}}{2\lambda}$$

$$Q_y = \frac{\omega_{y0}}{B_y} = \frac{\omega_{y0}}{2\lambda} \quad (3.59)$$

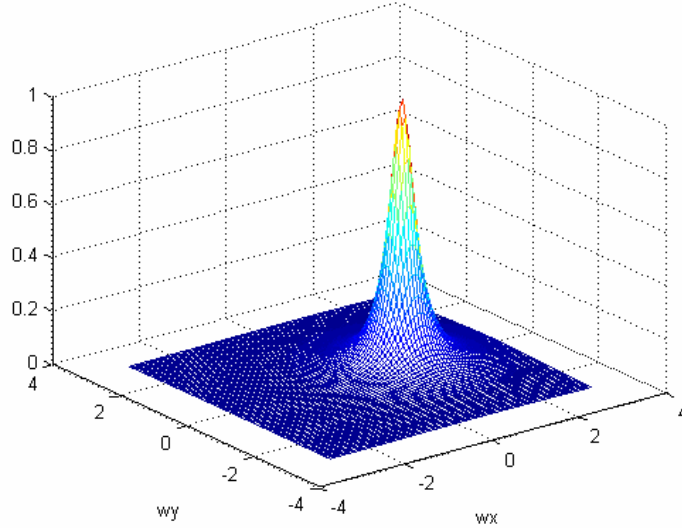
Toplam kalite faktörü  $Q_t$  süzgeç ayarının keskinliğini karakterize eder, ve şu şekilde gösterilir:

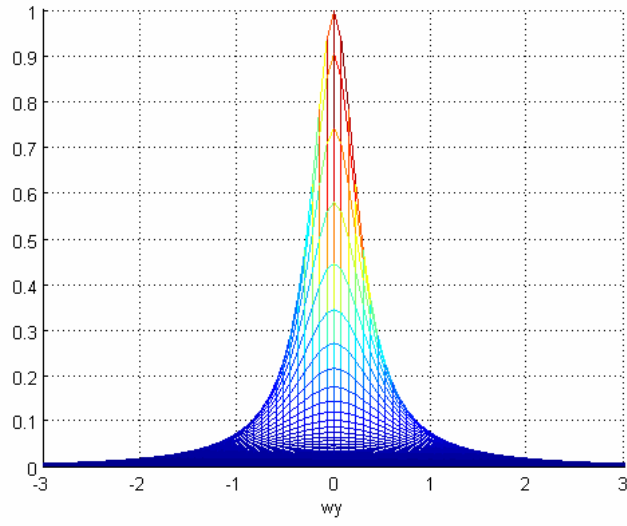
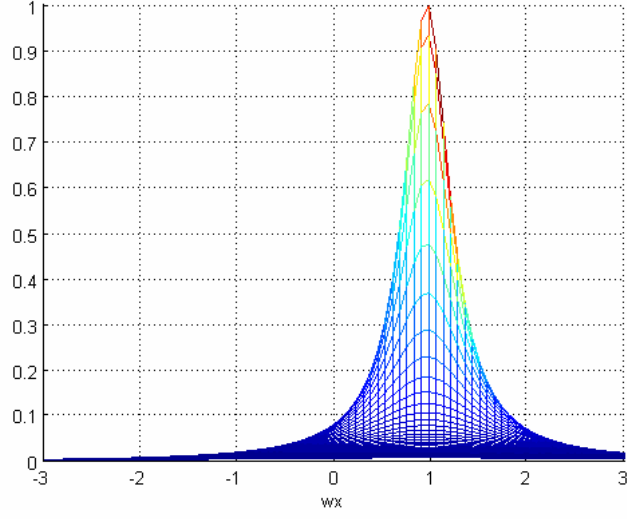
$$Q_t = \sqrt{Q_x^2 + Q_y^2} . \quad (3.60)$$

(3.59) daki değerler (3.60) eşitliğinde yerlerine konursa sonuç aşağıdaki gibi olur.

$$Q_t = \sqrt{\left(\frac{\omega_{x0}}{2\lambda}\right)^2 + \left(\frac{\omega_{y0}}{2\lambda}\right)^2} = \frac{\Omega_0}{2\lambda} \quad (3.61)$$

2-D Gabor tip süzgecin frekans yanıtının genlik değeri  $\Omega_0 = 1$ ,  $\lambda = 0.3$  ve  $\theta_0 = 0$  için şekil 3.8 de gösterilmiştir.



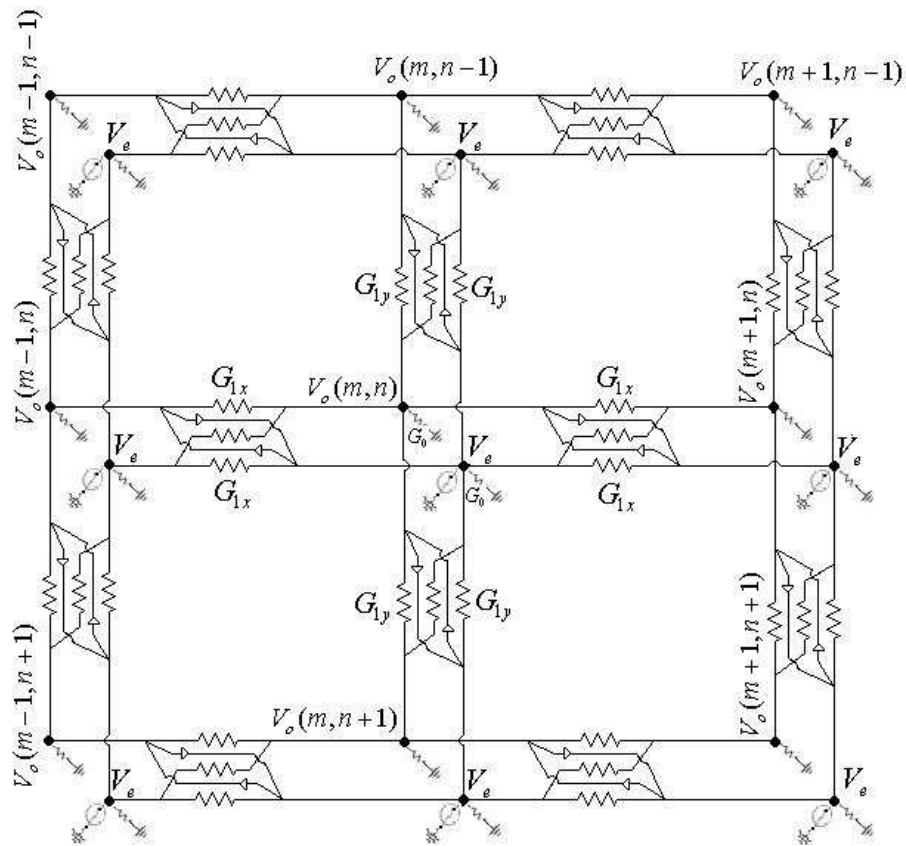


Şekil 3.8: HSA Gabor tip Süzgecin frekans yanıtı

## 4. 2 BOYUTLU HSA GABOR TİP SÜZGECİN DEVRE GERÇEKLEMELERİ

### 4.1 2 Boyutlu HSA Gabor Tip Süzgecin Analog Devre Gerçeklemesi

Shi [16] tarafından şekil 4.1 de gösterildiği gibi 2 boyutlu HSA Gabor tip süzgecin OTA ve dirençler ile gerçekleştirilebileceği önerilmiştir.



Şekil 4.1 2 Boyutlu HSA Gabor tip süzgecin OTA ve direnç kullanılarak analog gerçekleştirilmesi

Şekil 4.1 de verilen gerçekleştirme iki tane tek katmanlı HSA dizisinin birbirlerini yerel olarak etkilemesi ile elde edilen iki katmanlı bir yapıdır. Başka bir deyişle her hücre iki tane gerçek durum değişkenine sahiptir ( $v_e(.,.)$  ve  $v_o(.,.)$ ). Crouse [17]'de bant geçiren bir süzgecin oluşturula bilmesi için şablon boyutunun en az 5x5 olması gerektiğini göstermiştir fakat iki katmanlı HSA yapısı ile 3x3 boyutundaki şablonlar ile bu süzgecin gerçekleştirilebileceğini Shi [16,19] de göstermiştir. İlginç bir diğer yorum ise iki katmanlı yapıda katmanların sırası ile gerçek ve sanal kısımları göstermesidir [16,19].

Şekil 4.1 de gösterilen devrede dirençler iletkenlikleri ve üçgen bloklar OTA'lar ise kazançları ile adlandırılmışlardır. Elemanların değerleri şöyle verilmiştir:

$$G_0 = 4 + \lambda^2 - 2 \cos \omega_{x0} - 2 \cos \omega_{y0}$$

$$G_{1x} = \cos \omega_{x0} , G_{2x} = \sin \omega_{x0} , G_{1y} = \cos \omega_{y0} , G_{2y} = \sin \omega_{y0} \quad (4.1)$$

$\lambda$  (3.33) verilen denklemede gösterilen bir gerçek sayı sabitidir. Kolaylık için her düğümdeki kapasiteler ( $C=1$ ) alınmıştır. Giriş görüntüsündeki her bir pikselin ışık yoğunluğu  $u(m,n)$  akımı olarak gösterilir. Gabor tip süzgeç çıkışının gerçek ve sanal kısımlarını düğümlere ait  $v_e(m,n)$  ve  $v_o(m,n)$  kararlı durum voltajları ile ifade edilir. "e" ve "o" sırası ile çift ve teki ifade eder. Şimdi Shi tarafından verilen bu devrenin gerçeklik ve analizini yapacağız.

Şekil 4.1 de verilen devrede (m,n) düğümüne göre KCL yazarsak:

$$G_{1x} [V_e(m-1,n) - V_e(m,n)] + G_{1y} [V_e(m,n-1) - V_e(m,n)] +$$

$$G_{1x} [V_e(m+1,n) - V_e(m,n)] + G_{1y} [V_e(m,n+1) - V_e(m,n)] -$$

$$G_{2x} [V_o(m-1,n) - V_e(m,n)] - G_{2y} [V_o(m,n-1) - V_e(m,n)] +$$

$$G_{2x} [V_o(m+1,n) - V_e(m,n)] + G_{2y} [V_o(m,n+1) - V_e(m,n)] -$$

$$G_0 V_e(m,n) - C \frac{dV_e(m,n)}{dt} + \lambda^2 U(m,n) = 0 \quad (4.2)$$

$$\begin{aligned}
& G_{1x} [V_o(m-1, n) - V_o(m, n)] + G_{1y} [V_o(m, n-1) - V_o(m, n)] + \\
& G_{1x} [V_o(m+1, n) - V_o(m, n)] + G_{1y} [V_o(m, n+1) - V_o(m, n)] + \\
& G_{2x} [V_e(m-1, n) - V_o(m, n)] + G_{2y} [V_e(m, n-1) - V_o(m, n)] - \\
& G_{2x} [V_e(m+1, n) - V_o(m, n)] - G_{2y} [V_e(m, n+1) - V_o(m, n)] - \\
& G_0 V_e(m, n) - C \frac{dV_o(m, n)}{dt} = 0
\end{aligned} \tag{4.3}$$

Eleman deęerleri (4.2) ve (4.3)'te yerlerine konursa:

$$\begin{aligned}
\dot{V}_e(m, n) = & -(4 + \lambda^2) V_e(m, n) + \cos \omega_{x0} V_e(m-1, n) + \cos \omega_{y0} V_e(m, n-1) \\
& + \cos \omega_{x0} V_e(m+1, n) + \cos \omega_{y0} V_e(m, n+1) - \sin \omega_{x0} V_o(m-1, n) \\
& - \sin \omega_{y0} V_o(m, n-1) - \sin \omega_{x0} V_o(m+1, n) + \sin \omega_{y0} V_o(m, n+1) \\
& + \lambda^2 u(m, n)
\end{aligned} \tag{4.4}$$

$$\begin{aligned}
\dot{V}_o(m, n) = & -(4 + \lambda^2) V_o(m, n) + \cos \omega_{x0} V_o(m-1, n) + \cos \omega_{y0} V_o(m, n-1) \\
& + \cos \omega_{x0} V_o(m+1, n) + \cos \omega_{y0} V_o(m, n+1) + \sin \omega_{x0} V_e(m-1, n) \\
& + \sin \omega_{y0} V_e(m, n-1) - \sin \omega_{x0} V_e(m+1, n) - \sin \omega_{y0} V_e(m, n+1)
\end{aligned} \tag{4.5}$$

$$\frac{dV_e}{dt} = \dot{V}_e, \quad \frac{dV_o}{dt} = \dot{V}_o$$

Tanımlaması yapılırsa (4.4) ve (4.5) matris formatında şöyle gösterilir:

$$\begin{aligned}
\begin{bmatrix} \dot{V}_e(m, n) \\ \dot{V}_o(m, n) \end{bmatrix} = & \begin{bmatrix} \cos \omega_{x0} & -\sin \omega_{x0} \\ \sin \omega_{x0} & \cos \omega_{x0} \end{bmatrix} \begin{bmatrix} V_e(m-1, n) \\ V_o(m-1, n) \end{bmatrix} + \begin{bmatrix} \cos \omega_{y0} & -\sin \omega_{y0} \\ \sin \omega_{y0} & \cos \omega_{y0} \end{bmatrix} \begin{bmatrix} V_e(m, n-1) \\ V_o(m, n-1) \end{bmatrix} \\
& + \begin{bmatrix} \cos \omega_{x0} & \sin \omega_{x0} \\ -\sin \omega_{x0} & \cos \omega_{x0} \end{bmatrix} \begin{bmatrix} V_e(m+1, n) \\ V_o(m+1, n) \end{bmatrix} + \begin{bmatrix} \cos \omega_{y0} & \sin \omega_{y0} \\ -\sin \omega_{y0} & \cos \omega_{y0} \end{bmatrix} \begin{bmatrix} V_e(m, n+1) \\ V_o(m, n+1) \end{bmatrix} \\
& + \begin{bmatrix} -(4 + \lambda^2) & 0 \\ 0 & -(4 + \lambda^2) \end{bmatrix} \begin{bmatrix} V_e(m, n) \\ V_o(m, n) \end{bmatrix} + \begin{bmatrix} \lambda^2 u(m, n) \\ 0 \end{bmatrix}
\end{aligned} \tag{4.6}$$



Şimdi  $\dot{V}(m, n) = \dot{V}_e(m, n) + \dot{V}_o(m, n)$  tanımlanan ve (4.6) eşitliğinde iki gerçekteğerli durum değişkenlerini bir tane karmaşık değerli durum değişkenine birleştirildi. Karmaşık değerli tek katmanlı HSA'nın yerini tutan durum denklemi şöyle elde edilebilir:

$$\dot{V}(m, n) = -(4 + \lambda^2)V(m, n) + e^{j\omega_{x0}}V(m-1, n) + e^{-j\omega_{y0}}V(m, n+1) + e^{-j\omega_{x0}}V(m+1, n) + e^{j\omega_{y0}}V(m, n-1) + \lambda^2 u(m, n) \quad (4.7)$$

(4.7), (4.21) ve (4.22) ile karşılaştırılırsa, 2 boyutlu Gabor tip süzgeç için klonlanan şablonlar şöyle bulunur:

$$A = \begin{bmatrix} 0 & e^{-j\omega_{y0}} & 0 \\ e^{j\omega_{x0}} & -(4 + \lambda^2) & e^{-j\omega_{x0}} \\ 0 & e^{j\omega_{y0}} & 0 \end{bmatrix}; B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \lambda^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Devre parametreleri ile Gabor tip süzgeç parametreleri arasındaki bağıntı şu şekilde veriliyor:

$$\lambda^2 = G_0 + 2G_{1x} + 2G_{1y} - 4$$

$$\omega_{x0} = \arctan\left(\frac{G_{2x}}{G_{1x}}\right); \omega_{y0} = \arctan\left(\frac{G_{2y}}{G_{1y}}\right) \quad (4.8)$$

Bu nedenle süzgecin parametreleriiletkenler ve OTA ların değerleri ile elektronik olarak ayarlanabilir.

## 4.2 2 Boyutlu HSA Gabor Süzgecin Sayısal Gerçeklenmesi

$\dot{V}(m, n) = \dot{V}_e(m, n) + j\dot{V}_o(m, n)$  eşitliği ile verilen denklem, iki gerçekteğerli ve birbirine bağımlı durum Gabor tip süzgecin çıkışlarının yerini tutan  $\dot{V}_e(m, n), \dot{V}_o(m, n)$  eşitliklerine ayrılabilir.

$$\dot{v}_e(m,n) = \sum_{k,l=-1}^1 a_{k,l}^{ee} v_e(m+k,n+l) + \sum_{k,l=-1}^1 a_{k,l}^{eo} v_o(m+k,n+l) + b_e u(m,n)$$

$$\dot{v}_o(m,n) = \sum_{k,l=-1}^1 a_{k,l}^{oe} v_e(m+k,n+l) + \sum_{k,l=-1}^1 a_{k,l}^{oo} v_o(m+k,n+l)$$

Bu eşitliklerdeki değerler aşağıda verilmiştir.

$$A_{ee} = \begin{bmatrix} 0 & \cos \omega_{y0} & 0 \\ \cos \omega_{x0} & -(4 + \lambda^2) & \cos \omega_{x0} \\ 0 & \cos \omega_{y0} & 0 \end{bmatrix} \quad A_{oe} = \begin{bmatrix} 0 & -\sin \omega_{y0} & 0 \\ \sin \omega_{x0} & 0 & -\sin \omega_{x0} \\ 0 & \sin \omega_{y0} & 0 \end{bmatrix}$$

$$A_{eo} = \begin{bmatrix} 0 & \sin \omega_{y0} & 0 \\ -\sin \omega_{x0} & 0 & \sin \omega_{x0} \\ 0 & -\sin \omega_{y0} & 0 \end{bmatrix} \quad A_{oo} = \begin{bmatrix} 0 & \cos \omega_{y0} & 0 \\ \cos \omega_{x0} & -(4 + \lambda^2) & \cos \omega_{x0} \\ 0 & \cos \omega_{y0} & 0 \end{bmatrix}$$

$$b_e = \lambda^2$$

$A_{ee}, A_{eo}, A_{oe}, A_{oo}$  matrisleri iki katmanlı yapıdaki her bir katmandaki tek ve çift durumları göstermektedir.  $A_{ee}$  ve  $A_{oo}$  matrisleri aynı katmandaki ilişkileri,  $A_{eo}$  ve  $A_{oe}$  ise farklı katmanlardaki ilişkileri gösterirler. Bu matrisler arasındaki bağıntı ise şöyledir:

$$A_{oe} = -A_{eo}, A_{ee} = A_{oo}$$

Yukarıda verilen denklemlerde türev yerine Euler geri yaklaşıklığı uygulanarak sayısallaştırılır.

#### 4.2.1 2 Katmanlı 2 Boyutlu HSA GTS için Geri Euler Yaklaşıklığı

Geri Euler yaklaşıklığı kullanarak türev için yerine ifadesi yazılırsa:

$$V_e^{m,n}((q+1)T) = V_e^{m,n}(qT) + T \dot{V}_e^{m,n}(qT)$$

$$V_o^{m,n}((q+1)T) = V_o^{m,n}(qT) + T \dot{V}_o^{m,n}(qT)$$

q : Ayrık zaman indeksi,

T : Sayısal adım genizliği

m,n : Uzamsal indisler

$V_e^{m,n}((q+1)T)$  ve  $V_o^{m,n}((q+1)T)$  Gabor tip süzgecin çıkışları şu hale dönüşürler.

$$V_e^{m,n}((q+1)T) = \left[ 1 - a_{0,0}^{ee} T \right] V_e^{m,n}(qT) + T \left[ a_{-1,0}^{ee} V_e^{m-1,n}(qT) + a_{0,1}^{ee} V_e^{m,n+1}(qT) + a_{1,0}^{ee} V_e^{m+1,n}(qT) + a_{0,-1}^{ee} V_e^{m,n-1}(qT) + a_{-1,0}^{eo} V_o^{m-1,n}(qT) + a_{0,1}^{eo} V_o^{m,n+1}(qT) + a_{1,0}^{eo} V_o^{m+1,n}(qT) + a_{0,-1}^{eo} V_o^{m,n-1}(qT) + b_e u_o^{m,n} \right]$$

Şeklinde katsayılar hesaplanır. Eğer optimum adım genişliğini seçersek,  $T = \frac{1}{|a_{0,0}^{ee}|} = \frac{1}{|a_{0,0}^{eo}|} = \frac{1}{4 + \lambda^2}$

$V_e^{m,n}((q+1)T)$  ve  $V_o^{m,n}((q+1)T)$  değerlerini gösterelim.

$$V_e^{m,n}((q+1)T) = \frac{\cos \omega_{x0}}{4 + \lambda^2} V_e^{m-1,n}(qT) + \frac{\cos \omega_{y0}}{4 + \lambda^2} V_e^{m,n+1}(qT) + \frac{\cos \omega_{x0}}{4 + \lambda^2} V_e^{m+1,n}(qT) + \frac{\cos \omega_{y0}}{4 + \lambda^2} V_e^{m,n-1}(qT) - \frac{\sin \omega_{x0}}{4 + \lambda^2} V_e^{m-1,n}(qT) + \frac{\sin \omega_{y0}}{4 + \lambda^2} V_e^{m,n+1}(qT) + \frac{\sin \omega_{x0}}{4 + \lambda^2} V_e^{m+1,n}(qT) - \frac{\sin \omega_{y0}}{4 + \lambda^2} V_e^{m,n-1}(qT) + \frac{\lambda^2}{4 + \lambda^2} u^{m,n}$$

$$V_o^{m,n}((q+1)T) = \frac{\sin \omega_{x0}}{4 + \lambda^2} V_e^{m-1,n}(qT) - \frac{\sin \omega_{y0}}{4 + \lambda^2} V_e^{m,n+1}(qT) - \frac{\sin \omega_{x0}}{4 + \lambda^2} V_e^{m+1,n}(qT) + \frac{\sin \omega_{y0}}{4 + \lambda^2} V_e^{m,n-1}(qT) + \frac{\cos \omega_{x0}}{4 + \lambda^2} V_e^{m-1,n}(qT) + \frac{\cos \omega_{y0}}{4 + \lambda^2} V_e^{m,n+1}(qT) + \frac{\cos \omega_{x0}}{4 + \lambda^2} V_e^{m+1,n}(qT) - \frac{\cos \omega_{y0}}{4 + \lambda^2} V_e^{m,n-1}(qT)$$

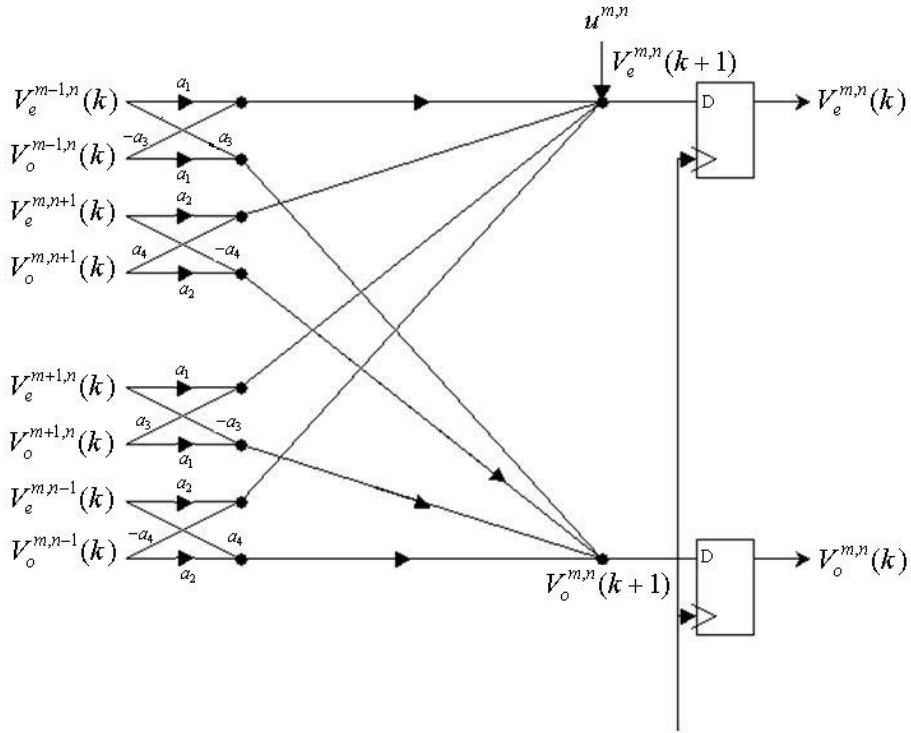
Yukarıdaki denklemlerin katsayılarını düzenlersek şu şekli alır.

$$a_1 = \frac{\cos \omega_{x0}}{4 + \lambda^2} , \quad a_2 = \frac{\cos \omega_{y0}}{4 + \lambda^2} , \quad a_3 = \frac{\sin \omega_{x0}}{4 + \lambda^2} , \quad a_4 = \frac{\sin \omega_{y0}}{4 + \lambda^2} , \quad b = \frac{\lambda^2}{4 + \lambda^2}$$

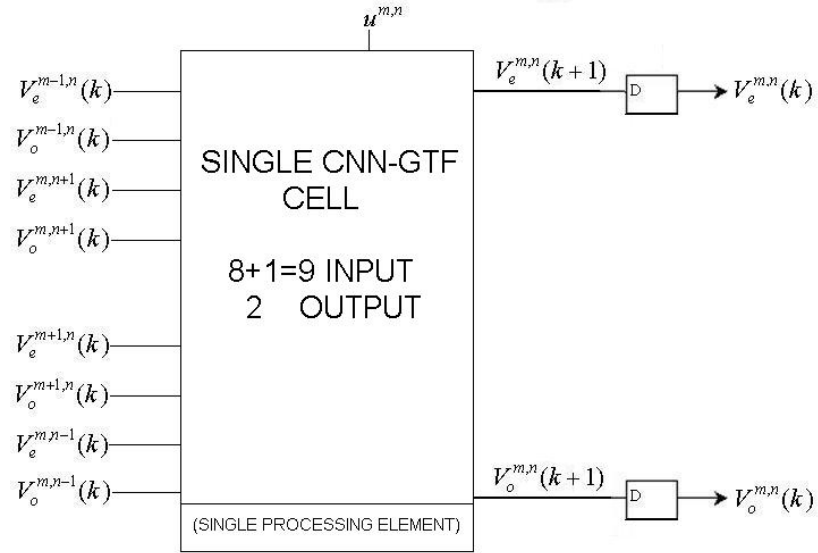
$$\begin{aligned}
V_e^{m,n}((q+1)T) &= a_1 V_e^{m-1,n}(qT) + a_2 V_e^{m,n+1}(qT) + a_1 V_e^{m+1,n}(qT) + a_2 V_e^{m,n-1}(qT) \\
&\quad - a_3 V_o^{m-1,n}(qT) + a_4 V_o^{m,n+1}(qT) + a_3 V_o^{m+1,n}(qT) - a_4 V_o^{m,n-1}(qT) \\
&\quad + b u^{m,n}(qT)
\end{aligned}$$

$$\begin{aligned}
V_o^{m,n}((q+1)T) &= a_3 V_e^{m-1,n}(qT) - a_4 V_e^{m,n+1}(qT) - a_3 V_e^{m+1,n}(qT) + a_4 V_e^{m,n-1}(qT) \\
&\quad + a_1 V_o^{m-1,n}(qT) + a_2 V_o^{m,n+1}(qT) + a_1 V_o^{m+1,n}(qT) + a_2 V_o^{m,n-1}(qT)
\end{aligned}$$

Elde edilen bağımlı yinelemeli yapıya sahip bu denklemler İki boyutlu HSA Gabor süzgecin tamamen sayısal devre olarak gerçekleştirilmesini sağlamaktadır. Bu devrenin blok yapısı aşağıdaki gibidir.



Şekil 4.2 : Sayısal 2 boyutlu HSA Gabor süzgecin akış diyagramı



Şekil 4.3 : 2 boyutlu HSA Gabor süzgeç işlem elemanı

## 5. FPGA'LERE GİRİŞ

Görüntü işleme (Gİ) uygulaması geliştiren kişilerin, gerçek zamanlı işlemlerde gereken yoğun işlemler için yüksek performanslı sistemlere ihtiyaç duymaktadırlar. Bu ihtiyaçlara ek olarak Gİ uygulamaları deneysel ve etkileşimli olma zorunluluğu vardır. Anlamı, geliştiriciler uygulamaları geliştirirken algoritmalarını değiştirebilmeli, ayarlayabilmeli ve yer değiştirebilmelidir.

Düşük seviyeli Gİ işlemlerinin (komşuluk işlemleri) doğasından dolayı, bu işlemleri paralel işlemlerle gerçekleştirmek performansı yükseltmektedir. Buna rağmen çok işlemcili Gİ sistemleri bu gerçeklemeler için tam bir çözüm olamamaktadırlar. Yüksek maliyet, paralel makineler için yazılım desteği ve kararlılık eksikliği: eğer görüntüler çekilip gerçek zamanlı olarak işlemcilere dağıtırlarsa, haberleşmeden dolayı oluşan masraflar oldukça fazladır.

Yüksek performanslı Gİ uygulamaları geliştirmenin ikinci bir yolu da Sayısal İşaret İşleyici (DSP) işlemciler kullanmaktır. DSP işlemciler temel olarak yazılım programlama desteği sayesinde geleneksel işlemcilere göre performansları geliştirilmiştir. Bu performans gelişmesi genel amaçlı bir işlemciden DSP algoritmaları gerçekleştirilebileceği bir yapıya dönüştürülmesi ile başarılmıştır. Örnek olarak DSP işlemciler bir çok veri yoluna sahiptir, bu özellik daha yüksek bant genişliği sağlamaktadır. Genel olarak bu yapı ardışık düzen (pipelined) kullanarak paralel işlemleri gerçekleştirmekte kullanılır. bu özelliklerine rağmen DSP işlemciler tek kontrol akışına bağımlılıklarından dolayı bir DSP işlemci paralel görevi tam olarak başaramaz.

Yukarıda belirtilen yazılım tabanlı çözümlere karşılık olarak, donanım çözümleri geliştirilmiştir. Uygulamaya özgül tümdevreler (ASIC) belirli bir algoritmayı tamamen isteğe uyarlanmış çözümlerdir. Bu çözüm tamamen bu algoritmanın çözümü içindir ve alan ve hız bakımından maksimum performansı sağlar. Bunlara rağmen yazılımlar kadar esnek olmaması

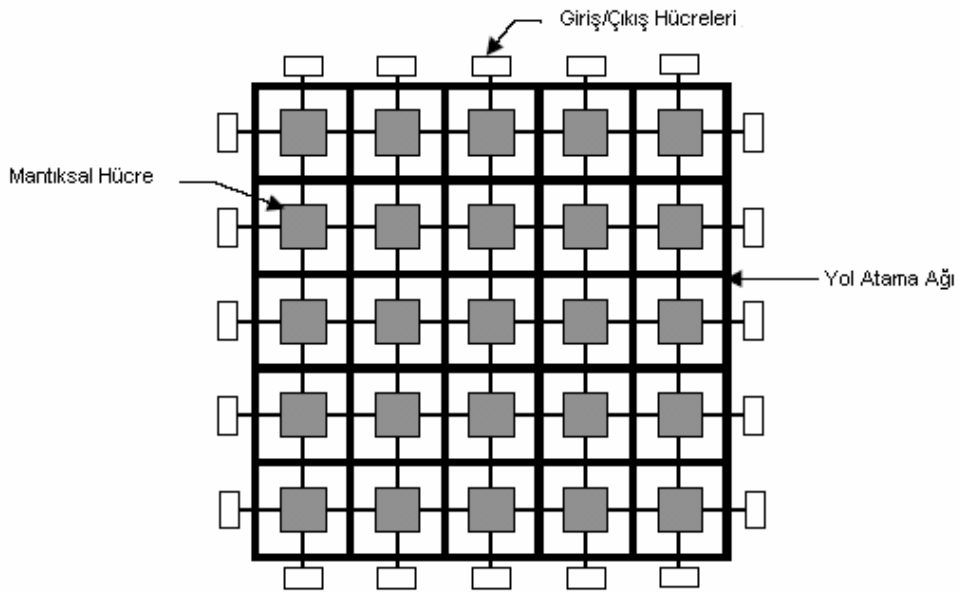
olumsuz yönüdür. Diğer olumsuz yönleri ise yüksek üretim maliyeti ve uzun üretim çevrimidir.

Tekrar ayarlanabilen donanım çözümlerinden, elektriksel olarak diğer bir algoritmanın tekrar programlanabileceği FPGA ler yüksek performans sağlarlar. Her ne kadar ilk FPGA ler basit mantık devreleri ve kontrol uygulamalarında kullanılsalar da, günümüzdeki FPGA ler milyon kapılarla ifade edilen bir yapıya kavuşmuşlardır. Üstelik tümleşik devre üretimindeki gelişmeler FPGA lerin çalışabileceği maksimum hızları arttırmıştır. Görüntü ve video işleme uygulamaları için geliştirilen günümüz FPGA leri 500 MHz lik bir bant genişliği ile çevresindeki diğer elemanlarla haberleşebilmektedir.

İlk önce FPGA lerin genel bir yapısı ve programlama yöntemleri ile başlayacağız. Daha sonraki bölümde ticari FPGA lerin yapıları incelenecek. Bu bölümün en son konusu ise uygulamanın yapıldığı devre incelenecektir.

## 5.1 FPGA'lere Genel Bakış

FPGA ler programlanabilen anahtarlar yardımı ile birbirine bağlanabilen mantık hücreleri dizisi içerir. Mantıksal hücreler paket pinlerine programlanabilen giriş çıkış hücreleri ile bağlıdır.



Şekil 5.1 FPGA'in genel yapısı

FPGAler şunlara göre ayrılabilirler;

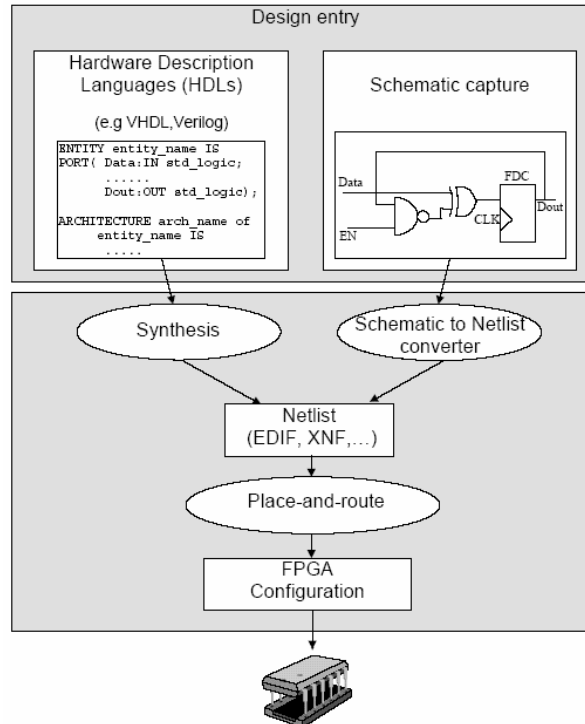
- mantıksal yapılarına göre,
- yol atamalarına göre.

FPGA lerde istenen durum programlanabilen elemanlarının (mantıksal hücreler, yol atama ağı, ve giriş çıkış hücreleri) ayarlanması ile oluşturulur. Bunlar programlanabilen anahtarların yardımı ile olur. Şu anki yüksek seviyeli FPGA ler mikroişlemciler gibi SRAM teknolojisi kullanılarak yapılırlar. Diğer teknolojiler ise antifuse ve FLASH yapılarıdır.

SRAM tabanlı FPGAler doğal olarak tekrar programlanabilirler. Bazı durumlarda çalışma anında belli bir bölgenin tekrar programlanması gerekir, bu yapı çalışma anındaki tekrar programlanmaya izin verir. Harici bir konfigürasyon hafıza kaynağı devreye güç verildiği zaman FPGA in programlanması için kullanılır. Konfigürasyon zamanı genel olarak 200 ms nin altındadır, konfigürasyon metoduna ve cihazın büyüklüğüne göre bu zaman değişmektedir.

## 5.2 FPGA Programlama

Tipik bir FPGA tasarım akış diyagramı şekil 5.2 de verilmiştir.



Şekil 5.2. FPGA tasarım çevriminin genel bir görünüşü



Yukarıdan aşağıya doğru bu diyagramdaki tasarım için gerekli adımları açıklayacağız, ilk olarak “design entry” den başlayacağız.

### 5.2.1 Tasarım (Design entry)

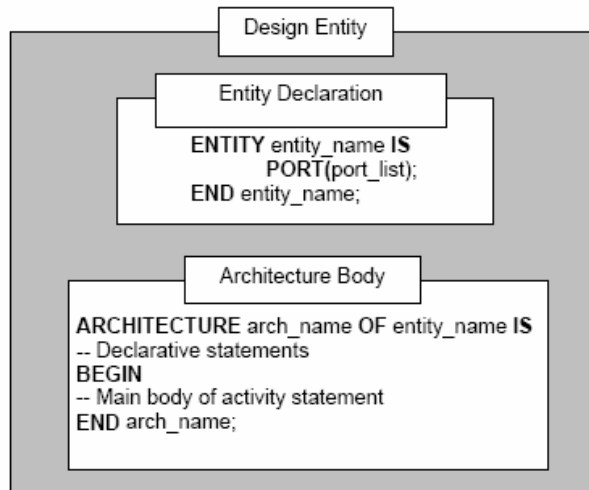
FGPA tasarımları ya şematik olarak yada donanım tanım dili kullanılarak metin halinde yapılır.

#### 5.2.1.1 Şematik tasarım (Schematic design entry)

Bu seçenekte, iki boyutlu bir arayüz programında mantıksal elemanları birbirine bağlayarak tasarım gerçekleşir. Tasarımda kullanılan bu elemanlar FPGA üreticisinin sağladığı kütüphane dosyalarında elde edilir. Üreticilere özel elemanlardan dolayı, özel bir donanım için yapılan şematik tasarımlar diğer donanımlar için uygun olmazlar. Şematik tasarım ile yapılan geliştirmelerde şematik tasarımdan net dosyalarını oluşturan bir editör kullanılır.

#### 5.2.1.2 Donanım Tanımlama Dilleri (Hardware Description Languages) (HDLs)

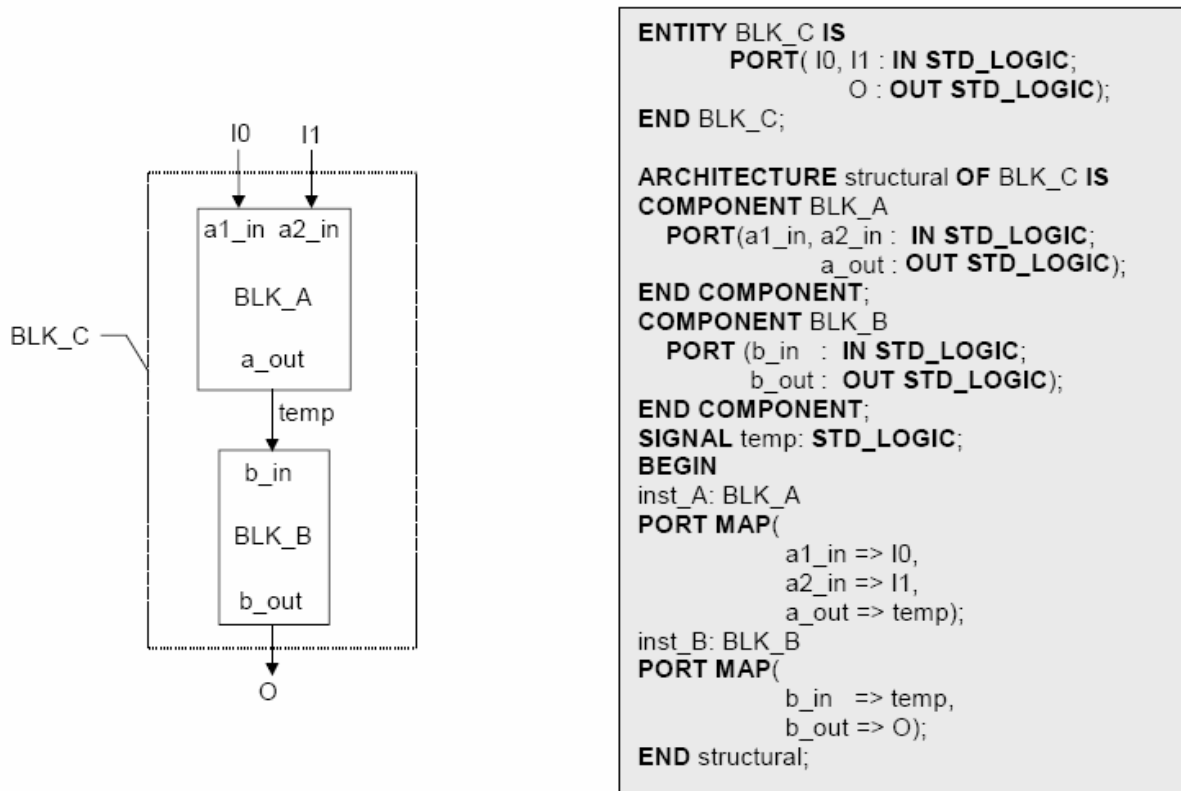
Birçok donanım tanımlama dili mevcuttur, örnek olarak VHDL ve Verilog. VHDL (Very-high-speed-integrated-circuit Hardware Description Language) en çok kullanılan HDL türüdür. VHDL dili donanım modeli için varlık tasarımı (design entity ) kavramını kullanır. Her varlık tasarımı (design entity) bir varlık bildirimini ve yapı bloğundan (şekil 1.3) oluşur.



Şekil 5.3 VHDL temel yapısı

Varlık bildirimini varlığın dış dünya bağlantılarını belirtir, tıpkı bir devrenin giriş ve çıkışları gibi. Yapı bloğu (Architecture body) ise giriş ve çıkışlar arasındaki ilişkinin nasıl olduğunu tanımlamaktadır.

Bir varlık birbiri ile ilişkili farklı yapılara sahip olabilir. Bunun bir yolu varlık tasarımının alt modüller ile yapılmasıdır. Alt modüllerin hepsi bir varlıktır ve sinyaller kullanılarak birbirlerine bağlanırlar. Şekil 1.4 de bu gibi bir yapı örnek olarak gösterilmiştir, 'BLK\_C' , 'BLK\_A' ve 'BLK\_B' gibi iki varlık ile oluşturulmuştur. Bu tip tanımlamalara yapısal tanımlamalar denir. 'BLK\_A' ve 'BLK\_B' yapılarını her ikisi de yapısal tanımlamalardır.



Şekil 5.4. VHDL yapısal tanımlama örneği

Çoğu durumda, varlık tasarımı yapısal olara ayrılmadan tanımlanır. Örnek olarak , eğer bir tasarım (IC - Integrated Circuits) tümdevreler ile yapılacaksa, tümdevrelerin iç yapılarını bilinmesine gerek yoktur. Bu gibi durumlarda modül tarafında gerçekleştirilen fonksiyonun tanımı gerekir. İç yapısına gerek duyulmamaktadır. Bu gibi tanımlamalara fonksiyonel (functional) ya da davranışsal (behavioural) tanımlama denir. Sıradaki örnek bir davranışsal tasarım olan özel veya (XOR2 ) fonksiyonudur.

```

ENTITY My_XOR2 IS
    PORT( I0, I1 : IN STD_LOGIC;
           O : OUT STD_LOGIC);
END My_XOR2;

ARCHITECTURE My_XOR2_arch OF My_XOR2 IS
BEGIN
    O <= (NOT(I0) AND I1) OR (NOT(I1) AND I0);
END My_XOR2_arch;

```

Şekil 5.5 Özel veya fonksiyonunun VHDL modeli

Kompleks yapılar bir fonksiyonun girişi olarak tanımlanamayabilirler. Geri dönüşümlü bir sistem, çıkış zamanının bir fonksiyonu olabilir. VHDL bu problemi çalıştırılabilen bir program formunda izin verilmiş davranışsal tanımlama yardımı ile çözer. Bu gibi durumlar için VHDL'e ekstra programlama yetenekleri eklenmiştir. Eklenenler sırası ile veri tipleri (integers, floating points, physical, enumeration, arrays), veri nesnelere (constants, variable and signals), ardışık ifadeler (if and case statements), alt programlar (procedures and functions) , paketlerdir. Aşağıdaki örnek kenar tetiklemeli D tipi bir flip-flop'un davranışsal VHDL gerçeğidir.

```

ENTITY My_DFF IS
    PORT(CLK, D : IN STD_LOGIC;
          Q : OUT STD_LOGIC);
END My_DFF;

ARCHITECTURE My_DFF_arch OF My_DFF IS
BEGIN
    reg: PROCESS(CLK)
        BEGIN
            -- detect the clock rising edge
            IF(CLK'event and CLK='1') THEN
                -- set the output 'Q' to the input value 'D'
                Q<=D;
            END IF;
        END PROCESS;
END My_DFF_arch;

```

Şekil 5.6 D tipi flip-flop

Kayıtçılar ve kayıtçılar arası mantık elemanları ile yapılan tanımlamalar RTL (Register Transfer Logic) tanımlamalardır. Bu tip tanımlamalar FGPA den bağımsız ve taşınabilirlerdir. VHDL kodu derleyici (synthesiser) tarafında derlenerek FPGA netlists dosyası oluşturulur. Şu anki derleyiciler toplama “+” ve çarpma “\*” gibi yüksek seviyeli tanımlamaları optimize ederek FPGA uygulaması için derleyebiliyorlar. VHDL kodu derleyici tarafında farklı FPGA modelleri için derlenebilmektedir.

### 5.2.2 Netlist Gösterimi

Netlist geliştirici tarafından geliştirilen tasarımın metin halinde gösterimidir. Tasarım fonksiyonunun yapısını ve tasarımda kullanılan FPGA üreticisine özel elemanları içermektedir. Çoğu üretici netlist gösteriminde FPGA üzerinde elemanları yerleşmelerinin gösterimine izin vermektedir. Netlist Electronic Data Format Interchange (EDIF) şeklinde standartlaştırılmıştır. Ayrıca üreticilerin kendi standartları da vardır.

### 5.2.3 Place-and-Route tools (PAR)

PAR aracı elemanları FPGA donanımının içinde yerleştirme ve yönlendirme işlerini yapar. Her FPGA üretici kendi PAR aracına sahiptir. Bu yerleştirme ve yönlendirme mekanizması netlist te hiyerarşi yapısına göre en iyi kapı düzenini oluşturmak için bazı optimizasyon işlemleri ve simülasyon analizleri yapar. Hücreler arası yönlendirme bu yerleştirme ve mevcut yönlendirme kaynakları tabanlı olarak gerçekleşir. Eğer PAR tasarım ile ilgili zaman kısıtlamaları bulursa, zaman yönlendirmesini otomatik olarak yapar.

PAR görevini başarılı bir şekilde yerine getirdiğinde, bitstream dosyası üretilebilir.

Bu bölümün sonunda FPGA ile gerçekleştirilmiş bir HSA simülatörünü inceleyeceğiz.

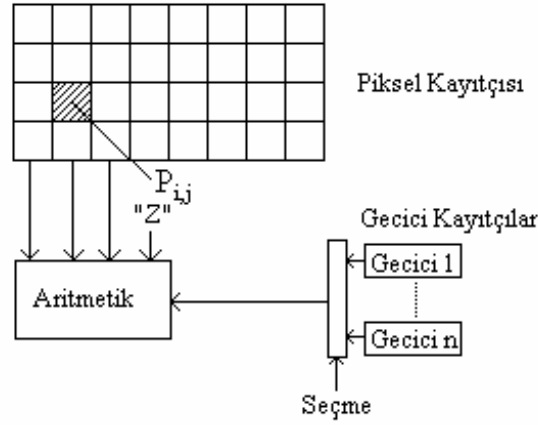
## 5.3 CASTLE ve Ardışık Düzen Yapısı İle Gerçeklenen HSA Simülatörü

Farklı yapılarda HSA gerçeklemeleri mevcuttur, bunlardan iki tanesi CNN-UM ve sayısal olarak ta CASTLE. Bu sayısal gerçekleştirme analog sistem ile aynı sonuçları verir, sadece hız farkı vardır. Eğer geçici hesaplamalar önemli ise hızdaki bu kayıp büyük bir eksikliklerdir. CASTLE pipe-line tekniğini kullanarak, sayısal gerçekleştirme söz konusu edilen hızı arttırmıştır.

Burada bu çözümün alan, hız ve zaman analizi yapılacaktır. Bu aritmetik çekirdek VIRTEX FPGA geliştirme seti üzerinde test ve benzetimi yapılmıştır.

### 5.3.1 Orijinal CASTLE Mimarisi

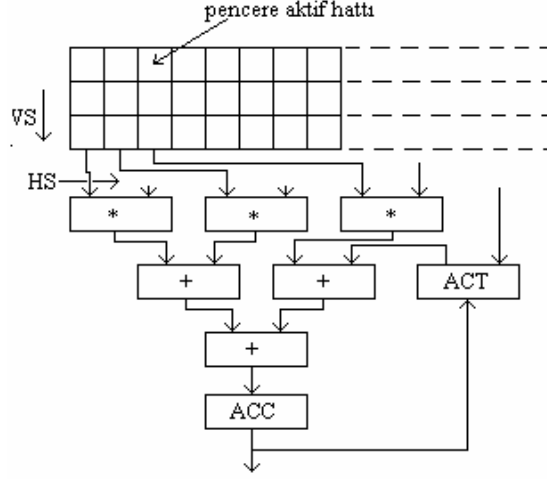
CASTLE HSA\_EM (CNN-UM) sayısal bir kopyasıdır. Analog HSA hücresi geri Euler metodu kullanılarak fark denklemi haline getirilmiştir. Orijinal CASTLE mimarisi şekil 5.7 görülebilir. Bu mimari bir aritmetik, piksel ve şablon hafıza birimlerinde oluşur. Bu işlem ünitesi mandal tipi kayıtçılarla oluşturulmuş ACC ve ACT oluşmaktadır. ACC ve ACT kayıtçıları sırası ile akümülatör ve geçici akümülatörlerdir.



Şekil 5.7 : CASTLE mimarisi

Piksellerin ve şablonların komşulukları için üç adet çarpıcı kullanılmıştır. CASTLE ın bu aritmetik çekirdeği şekil 5.8. görülebilir.

3x3 boyutundaki kare pencere piksel hafızasında özel bir yol üzerinde hareket eder. Farklı şablonlar ve eşik değerleri için bu dokuz değer ortadaki piksel değerinin hesaplanmasında kullanılır. Bu aritmetik birimin gecikmesi çok uzundur, yaklaşık olarak 10 ns (6+2+2). Bu gecikme limit saat frekansı ve üç katmandaki gecikmelerin toplamıdır. Daha detaylı bilgiler için tablo 5.1 e bakılabilir. Pencere hareketinin yolu şunları izler: Aktif pencere hattı yatay yönde bir piksel ve daha sonra dikey yönde üç pikseldir. Bu işlem n zaman için n uzunluğundaki piksel satırı için tekrarlanır.



Şekil 5.8. Orijinal CASTLE aritmetik çekirdeği

	Gecikme [ $ns$ ]	Boyut [ $\mu m^2$ ]
Kayıtçı	□ 0.5	256
Tam Toplayıcı	□ 0.5	457
LACA	□ 2	3476.16
6 bit doğruluklu çarpıcı	□ 6	110120
6 bit doğruluklu toplayıcı	□ 2	28188
Çoklayıcı	□ 0.5	705

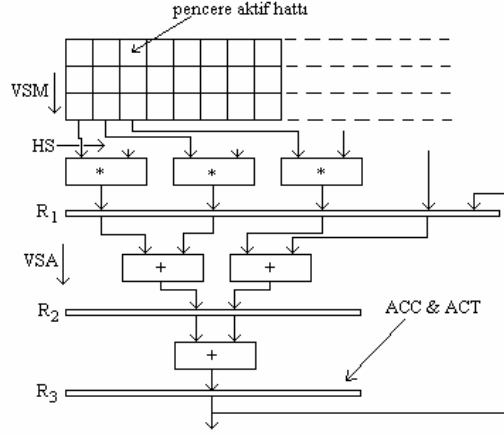
Tablo 5.1: Farklı gecikmeler

### 5.3.2 Ardışık (Pipe-line) yapılı CASTLE Mimarisi

Orijinal CASTLE aritmetik çekirdeği geçici kayıtçılar ile daha etkili hale gelebilir [20].

$R_1, R_2$  ve  $R_3$  kayıtçı grubu köle sahip yapısındadır. Orijinal yapının ACC ve ACT kayıtçıları  $R_3$  grubunda birleştirilmiştir. Sonuç olarak işlem hızı önemli şekilde artar.

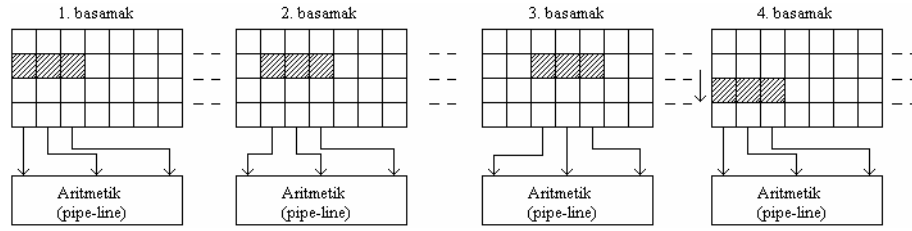
Pipe-line çözümündeki pencere hareketi orijinal CASTLE mimarisindekinden farklıdır. Aktif hattın bir piksel düşey yönde hareketinden sonra aynı anda aktif hat üç defa yatay yönde ve hesaplamının ara sonucunda bir defa düşey yönde kayar (şekil 5.9). Bu işlem  $n/3$  zaman aralığıyla tekrar eder.



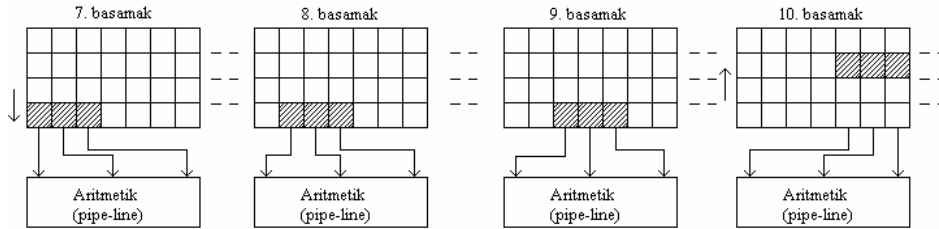
Şekil 5.9. Ardışık düzenli CASTLE mimarisi

Şekil 5.10 ve 5.11 yukarıdaki işlemleri daha detaylı bir şekilde izah etmektedir. Pencerenin aktif hattının pozisyonu bu şekillerin farklı yerlerindedir.

Pipe-line aritmetik birimindeki ara sonuçların düşey yönde kaymasını daha detaylı incelenirse, ilk işlenen sonucun  $R_3$  kayıtlarına girmesi 12. (9+3) adımda olacaktır.



Şekil 5.10 : Ardışık düzenli CASTLE mimarisinde hafıza kullanımı (1)



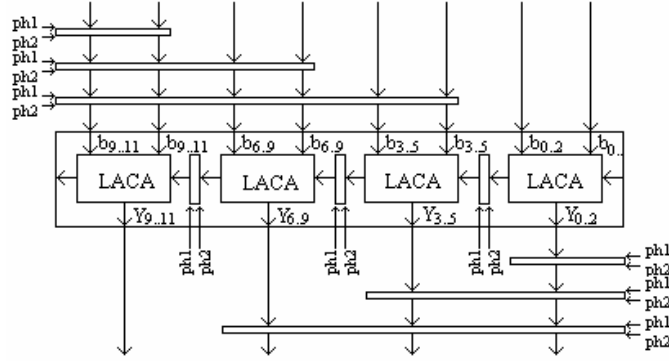
Şekil 5.11 : Ardışık düzenli CASTLE mimarisinde hafıza kullanımı (2)

### 5.3.3 Bloklar içinde Ardışıklık (Pipelining)

Çarpıcıların gecikmeleri 6 ns (tablo 5.2) olduğundan dolayı en yüksek işlem hızı yaklaşık olarak 170 MHz dir. Eğer çarpıcı ve toplayıcılarda da pipelining tekniği kullanılırsa bu işlem

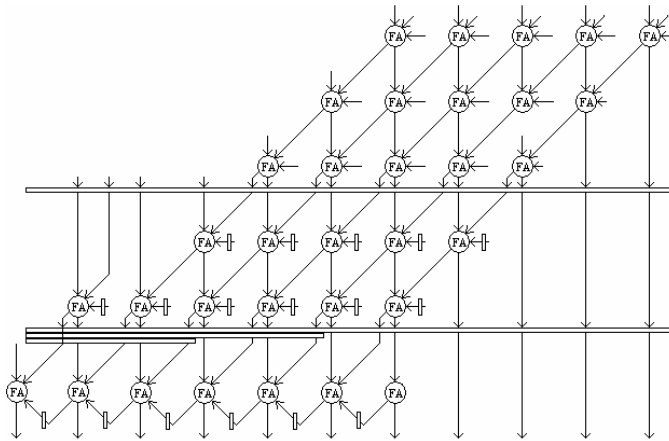
hızı arttırılabilir. Hafıza sistemindeki metot “k” katmanlı ( $k>3$ ) pipe-line da kullanılabilir. Bundan dolayı işlem hızı önemli ölçüde artar.

CNN-UM (CASTLE) sayısal tasarımında Baugh-Wooley çarpıcılar kullanılmıştır. Orijinal Baugh-Wooley çarpıcılarının gecikme yaklaşık olarak 6 ns dir (12\* tam toplayıcı). Aritmetik birimin çarpıcı ve toplayıcıları bazı geçici kayıtçılarla tamamlanabilmiştir (Şekil 8,9). Orijinal toplayıcının gecikmesi yaklaşık olarak 2 ns dir ( $0.35\mu\text{m}$  teknolojisi). Eğer iki LACA arasına bir geçici kayıtçı koyarsak bu gecikme süresi 1 ns ye kadar düşürülebilir.



Şekil 5.12 : Ardışık düzenli toplayıcı

Eğer şekil 5.13 da gösterildiği gibi orijinal diziler arasında geçici kayıtçılar konulursa çarpıcıların hızları önemli ölçüde arttırılabilir.



Şekil 5.13 : Ardışık düzenli “Baugh-Wooley” çarpıcısı

Bu durumda gecikme yaklaşık olarak 1 ns dir (üç tam toplayıcılı durum için en iyisidir.). Pipe-line seviye sayısı “k” dır. Hafıza yönetim tekniği şekil 5.13 da görülebilir.



### 5.3.4 Kayıtçı Grubuna Giriş Nasıl Sağlanır?

Geliştirilen CASTLE mimarisi yaklaşık olarak 1 GHz hızındaki kayıtçılarla çalışır (tablo2). Buradaki en büyük problem, geliştirilen HSA yongası hafıza kayıtçılarında görüntü parçalarının nasıl yazılacağı ve nasıl okunacağıdır. Eğer bir aritmetik birim birden fazla bağımsız yerel hafızalara sahip ise, bu problem için çok güzel bir çözümdür. Piksel hafızasının işlem sıklığı yaklaşık olarak 200 MHz ve aritmetik birimin gecikmesi ise yukarıda 1 ns (1 GHz)dir, bundan dolayı bağımsız yerel hafızaların sayısı 5 olabilir. O halde aritmetik çekirdekte beş yerel hafıza kullanılabilir. Tamamlanmış yeni CASTLE işlemcisi şekil 11 de görülebilir. Bu işlemci beş yerel hafıza, kontrol ve zamanlama birimleri ve bir aritmetik birimden oluşur (CASTLE 5\*1 dizi).orijinal görüntü beş parçaya atılır. Bu görüntü parçaları ( $görüntü_{1-5}$ ) yerel hafızalara yerleştirilir ( $Locmem_{1-5}$ ).Orijinal CASTLE ın işlem hızı 5\*1 CASTLE ın işlem hızına eşittir fakat yeni CASTLE ın dizi alanı daha küçüktür.

Orijinal CASTLE (şekil 1) işlemcilerden oluşturulmuş eski CASTLE dizi işlemcileri (Şekil 12). Yeni ve eski CASTLE ları G/Ç hızlarına göre karşılaştırıldığında bir fark yoktur, dizi işlemcinin silikon alanında önemli bir düşüş vardır. Yeni CASTLE dizi şekil 13 de gösterilmiştir.

### 5.3.5 SONUÇ

Bundan önceki bölümlerde XILINX geliştirme kiti ve VHDL benzetim aracıyla benzetimi kurulan CASTLE aritmetik birimlerinin farklı sonuçları işlenmişti. Farklı benzetim ve hesaplama parametreleri tablo ikide gösterilmektedir. Gerçeklemelerden ilki VIRTEX FPGA ( $0.18\mu m$  CMOS) diğeri ise tamamen ısmarlama yapılan ASIC ( $0.35\mu m$  CMOS) uygulamasıdır. (piksel ve şablonların bir derinlikleri 6 bittir.) [22]

	CASTLE(3x3)	CASTLE(3x3) pipeline	CASTLE(3x3) pipeline (toplayıcılar,çarpıcılar.)
Gecikme Formülü	$t_{orijinal}$	$t_{\text{çarpıcı}} + t_{\text{kayıtçı}}$	$3xt_{\text{tam-toplayıcı}}$
Gecikme [ns]	□ 10	□ 6	□ 1.5
Alan [ $mm^2$ ]	5.8382	5.8513	5.944952
En üst saat frekansı (VIRTEX300, $0.18\mu m$ )	46 MHz	196.928 MHz	313.972 MHz
En yüksek saat frekansı (silikon üzerinde, $0.35\mu m$ )	□ 100 MHz	□ 170 MHz	□ 1 GHz
Enerji yitimi [W]	< 0.3	< 0.5	< 3

Tablo 5.2 : Gecikme zamanları ve çip büyüklükleri

## 6. İKİLİK SAYI SİSTEMİ VE SABİT NOKTALI ARİTMETİK

Tam sayı değerlerini göstermek için ikilik sayı sistemi iki farklı biçimde kullanılabilir. Eğer yalnızca pozitif ve sıfır tamsayıları gösteriliyorsa, bu biçimde kullanılan ikilik sayı sistemine işaretli ikilik sayı sistemi denir. İkilik sayı sisteminin negatif sayıları da kullanmasına, işaretli ikilik sayı sistemi denir.

### 6.1. İkilik Sayı Sisteminde İfade Edilen Bir Tamsayının Onluk Sistemdeki Karşılığı

İkilik sayı sisteminde ifade edilen bir sayının onluk sayı sistemindeki karşılığını hesaplamak için en sağdan başlanarak bütün basamaklar tek tek ikinin artan üstleri ile çarpılır. Örneğin,

$$(1011)_2 = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = 11$$

İkilik sayı sisteminde yazılan bir sayının en solundaki bit, yukarıdaki örnekten de görüldüğü gibi en yüksek sayısal değeri katar. Bu bite en yüksek anlamlı bit denir.

İkilik sayı sisteminde ifade edilen bir sayının en sağındaki bit, yine yukarıdaki örnekten de görüldüğü gibi en düşük sayısal değeri katar. Bu bite en düşük anlamlı bit denir.

01011001 sayısı için en değerlikli bit '0', en düşük değerlikli bit ise '1' dir.

### 6.2 İkilik Sistemde İfade Edilen Bir Sayının Bir Tümleneni

Sayıyı ifade eden bütün değerlerin tümleninin alınmasıyla elde edilir. Bir sayının bire tümleneni tekrar 1 e tümlen işleme sokarsak sayının kendisini elde ederiz.

$$10101100 \rightarrow 01010011$$

### 6.3 İkilik Sistemde İfade Edilen Bir Sayının İkiye Tümleyeni

Bir sayının bire tümleyenine 1 eklersek o sayının ikiye tümleyenini bulmuş oluruz. İkiye tümleyen hesaplanmasının pratik bir yolu vardır, sayının sağından başlayarak ilk 1'e kadar (ilk 1 dahil) sayının kendisi yazılır, o 1 den sonra ise her bir bitin tümleyeni yazılır. Bu şekilde sayının ikiye tümleyenini yazmış oluruz.

$$10110110 \rightarrow 01001001 + 00000001 = 01001010$$

Aynı şekilde bir sayının peş peşe iki kere ikiye tümleyenini alırsak tekrar sayının kendisini buluruz.

### 6.4 Negatif Bir Sayının İkilik Sistemdeki Gösterimi

İşaret biti "0" ise pozitif, "1" e eşit ise negatiftir denir. İşaret biti en ağırlıklı bitin solunda yer alır. İkilik sayı sisteminde bir negatif sayı aynı değerdeki pozitif sayının ikiye tümleyenidir.

İşaretsiz 10011101 sayısının onluk sayı sisteminde hangi sayıya karşılık geldiğini bulalım. İlk önce işaret bitine bakılır, 1 olduğu için sayı negatiftir denir. Negatif sayı aynı değerlikli pozitif sayının ikiye tümleyenidir. Bu nedenle bu sayının tekrar ikiye tümleyenini alırsak bu sayının değerini bulmuş oluruz.

$$10011101 \rightarrow 01100010 + 00000001 = 01100011 = 99$$

10011101 bu sayı demek ki -99 sayısını göstermektedir.

### 6.5 Ondalık Sayıların İkilik Sayı Sisteminde Gösterimleri

Sabit noktalı gösterim programcı tarafından veri uzunluğu içerisindeki iki bit arasına koyduğu sanal nokta ile ifade edilmektedir.

Q[QI].[QF] şeklinde ifade edilen sayıda QI sayısının tam kısmını QF ise kesirli kısmını gösterir. QI+QF toplamı kelime uzunluğunu verir. Geleneksel işlemciler için kelime uzunluğu 8, 16, 32, 64 dür. Örnek olarak Q3.5 sayısı 3 ü tam, 5 i kesirli olmak üzere 8 bitlik bir değerdir.

İşlem yapılacak işaretli (pozitif) kesirli sayının, tam kısmının gösterimi için kaç bit gerektiği (6.1) ile hesaplanır.

$$0 \leq \alpha \leq 2^{QI}$$

$$QI = \text{üste\_yuvarla}(\log_2(\text{abs}(\alpha))) \quad (6.1)$$

$\alpha = 5.4321$  değeri için (6.1) i kullanalım.

$$QI = \text{üste\_yuvarla}(\log_2(\text{abs}(5))) = \text{üste\_yuvarla}(2.3219) = 3$$

$$5.4321 \leq 2^3 = 8$$

$\alpha$  sayısının tam kısmını gösterebilmek için 3 bit kullanılması gerektiği gösterilmiştir.

Eğer işaretli bir sayı ise QI'nin içinde işaret bitide bulunacaktır. (6.2) ile işaretli kesirli bir sayının tam kısmının kaç bit olacağı gösterilmiştir.

$$-2^{QI-1} \leq \alpha < 2^{QI-1}$$

$$QI - 1 = \text{üste\_yuvarla}(\log_2(\max(\text{abs}[\alpha_{\max}, \alpha_{\min}]))) \quad (6.2)$$

$\alpha = -5.4321$  değeri için (6.2) i kullanalım.

$$QI - 1 = \text{üste\_yuvarla}(\log_2(\max(\text{abs}[-5]))) = \text{üste\_yuvarla}(\log_2(5))$$

$$QI = \text{üste\_yuvarla}(\log_2(5)) + 1 = \text{üste\_yuvarla}(2.3219) + 1 = 4$$

$$QI = 4$$

Sabit noktalı sayılarda çözünürlük ondalıklı kısım ile verilmektedir. Sabit noktalı bir sayının çözünürlüğü (6.3) ile verilir.

$$\varepsilon = \frac{1}{2^{QF}} \quad (6.3)$$

İstenen çözünürlük için gerekli QF değeri (6.4) ile hesaplanır.

$$QF = \text{üste\_yuvarla} \left( \log_2 \left( \frac{1}{\varepsilon} \right) \right) \quad (6.4)$$

$\alpha = -5.4321$  değeri için gerekli QF değeri;

$$\varepsilon \leq 0.0001$$

$$QF = \text{üste\_yuvarla} \left( \log_2 \left( \frac{1}{0.0001} \right) \right)$$

$$QF = \text{üste\_yuvarla} (\log_2(10000)) = \text{üste\_yuvarla} (13.288) = 14 \quad \text{olarak bulunur.}$$

Ondalıklı ve tam kısım için gereken bit sayılarının toplamı QI+QF kelime uzunluğunu vermektedir. İfade edilebilecek sayıların sınırı ise (6.5) ile gösterilmiştir.

$$-2^{QI-1} \leq \alpha < 2^{QI-1} - 2^{QF} \quad (6.5)$$

(6.5) e göre  $\alpha = -5.4321$  sayısı için gerekli kelime uzunluğu  $WL = QI + QF = 4 + 14 = 18$  olur.

Kayar noktalı sayıya karşılık gelen sabit noktalı (tamsayı) (6.6) ifadesi ile hesaplanır.

$$FxdPt = \text{alta\_yuvarla}(FltPt \times 2^{QF}) \quad (6.6)$$

$\alpha = -5.4321$  değeri için QF=14 değeri hesaplanmıştır.  $\alpha$  sayısının tamsayı (sabit nokta) gösterimi :

$$\alpha_{FxdPt} = \text{alta\_yuvarla}(\alpha_{FltPt} \times 2^{QF}) = \text{alta\_yuvarla}(-5.4321 \times 2^{14})$$

$$\alpha_{FxdPt} = \text{alta\_yuvarla}(-5.4321 \times 16384) = \text{alta\_yuvarla}(-88999,5264) = -88999 \quad \text{bulunur.}$$

## 6.6 Sekiz Bitlik Sayılarla Matematiksel İşlemler

$\pm 1(\alpha)$  ve  $\pm 2(\beta)$  değerleri arasında değişen iki değişken alınsın. Kelime uzunluğu 8-bit olması durumunda  $\alpha$  ve  $\beta$  sayılarının sabit nokta gösterimi için, Q1.7, Q2.6 ihtiyaç vardır.

### 6.6.1 Q1.7 Gösterimi

Q1.7 gösteriminde -1 ile 0.9921875 (1-1/128) arasında 0.0078125 artımlarla olan sayılar ifade edilebilir. Q1.7 gösteriminin basamak ağırlıkları tablo 6.1 de verilmiştir.  $\alpha$  değeri Q1.7 formatında gösterilir.

s.	x	x	x	x	x	x	x
-1	1/2	1/4	1/8	1/16	1/32	1/64	1/128

Tablo 6.1 Q1.7 gösteriminin basamak değerleri

### 6.6.2 Q2.6 Gösterimi

Sekiz bitlik Q2.6 gösterimi ile -2 ile 1.984375 (2-1/64) arasındaki 0.015625 artımlı sayıları ifade edilebilir. Q2.6 gösteriminin basamak ağırlıkları tablo 6.2 de verilmiştir.

s	x.	x	x	x	x	x	x
-2	1	1/2	1/4	1/8	1/16	1/32	1/64

Tablo 6.2 Q2.6 gösteriminin basamak değerleri

### 6.6.3 Q1.7xQ2.6=Q3.13 Çarpma İşlemi

WL bitlik kelime uzunluğuna sahip iki tam sayının çarpımında sonuç  $2xWL$  boyutunda olur. Eğer tamsayılar sabit noktalı değer içeriyorsa, sonucun tam ve ondalıklı kesiminin kelime uzunlukları çarpanlarının tam ve ondalıklı kesimlerinin kelime uzunları toplamıdır (6.7).

$$\begin{aligned}
 QI_{\text{Sonuç}} &= QI_{\text{Çarpan}} + QI_{\text{Çarpılan}} \\
 QF_{\text{Sonuç}} &= QF_{\text{Çarpan}} + QF_{\text{Çarpılan}}
 \end{aligned}
 \tag{6.7}$$

(6.7) e göre Q1.7 ve Q2.6 formatındaki iki sayının çarpımında sonuç Q3.113 formatında olacaktır. Q3.13 gösterimi -4 ile 3.9998779296875 (4-1/8192) sayıları arasında

0.0001220703125 artımlı sayıları ifade eder. Q3.13 gösteriminin basamak ağırlıkları tablo 6.3 de verilmiştir.

s	x	x.	x	x	x	x	x	x	x	x	x	x	x	x	x
-4	2	1	1/2	1/4	x	x	x	x	x	x	x	x	x	x	1/8192

Tablo 6.3 Q3.13 gösteriminin basamak değerleri

Q3.13 formatındaki sonucun 8 bit olarak ifade edilmesi gerekir, nedeni her çarpma işleminde sayının devamlı daha büyük bir bellek alanı istemesidir. 8 bit olarak göstermek için sonucun ondalıklı kısmından 8 bit atılır ve sonuç Q3.5 formatında saklanır (tablo 6.4).

s	x	x.	x	x	x	x	x	x	x	x	x	x	x	x	x
-4	2	1	1/2	1/4	x	x	x	x	x	x	x	x	x	x	1/8192

Tablo 6.4 Q3.5 gösteriminin basamak değerleri (gri kısım atılır)

#### 6.6.4 Q1.7+Q2.6=Q2.6 Toplama İşlemi

Toplama tıpkı tamsayı toplama gibidir fakat sabit noktalı toplamda noktanın yeri aynı hizada olmasına ve taşmaya dikkat edilmelidir.

$$\begin{array}{r} |s|x.|x|x|x|x|x| \\ +|s|x|x.|x|x|x|x|x| \end{array}$$

Q1.7 formatındaki sayı noktanın üst üste gelmesi için bir basamak sağa kaydırılır ve önüne işaret bitinin aynısı yazılır.

$$\begin{array}{r} |s|x.|x|x|x|x|x| \\ +|s|x|x.|x|x|x|x|x| \\ |c|s|x|x.|x|x|x|x|x| \end{array}$$

Eğer kelime uzunluğundan daha büyük bir sonuç bulunursa (taşma-overflow) elde biti kontrol edilir.



### 6.6.5 Gerçek Sayılar Kullanılan Bir Örnek

Kelime uzunluğu 8 bit,  $|\alpha| \leq 1.8$ ,  $|\beta| < 1$  ve  $|\chi| \leq 2.8$  değerleri arasında olan değişkenler olsun.  $\alpha = 1.667$ ,  $\beta = -0.75$  ve  $\chi = 2.6$  alınarak maksimum çözünürlüğü hesaplayalım:

$$QI_{\alpha} = \text{üste\_yuvarla}(\log_2(\max(\alpha))) + 1 = \text{üste\_yuvarla}(\log_2(1.8)) + 1 = 2$$

$$QF_{\alpha} = WL - QI_{\alpha} = 8 - 2 = 6$$

$$\varepsilon_{\alpha} = \frac{1}{2^{QF_{\alpha}}} = \frac{1}{2^6} = \frac{1}{64} = 0.015625$$

$$-116_{10}(Q2.6) \leq \alpha_{FxdPt} \leq 115_{10}(Q2.6)$$

$$QI_{\beta} = \text{üste\_yuvarla}(\log_2(\max(\beta))) + 1 = \text{üste\_yuvarla}(\log_2(1)) + 1 = 1$$

$$QF_{\beta} = WL - QI_{\beta} = 8 - 1 = 7$$

$$\varepsilon_{\beta} = \frac{1}{2^{QF_{\beta}}} = \frac{1}{2^7} = \frac{1}{128} = 0.0078125$$

$$\beta_{FxdPt} = -0.75 \times 2^7 = -96_{10}(Q1.7)$$

$$QI_{\chi} = \text{üste\_yuvarla}(\log_2(\max(\chi))) + 1 = \text{üste\_yuvarla}(\log_2(3)) + 1 = 3$$

$$QF_{\chi} = WL - QI_{\chi} = 8 - 3 = 5$$

$$\varepsilon_{\chi} = \frac{1}{2^{QF_{\chi}}} = \frac{1}{2^5} = \frac{1}{32} = 0.03125$$

$$-90_{10}(Q3.5) \leq \chi_{FxdPt} \leq 89_{10}(Q3.5)$$

$$\chi_{FxdPt} = 2.6 \times 2^5 = 83_{10}(Q3.5)$$

$$(\alpha \times \beta) + \chi = (1.667 \times -0.75) + 2.6 = -1.25025 + 2.6 = 1.34975$$

$(\alpha \times \beta)$  elemanının hesabı :

$$\alpha_{FxdPt} \times \beta_{FxdPt} = 106_{10}(Q2.6) \times -96_{10}(Q1.7) = -10176_{10}(Q3.13) = -1.2421875$$

Sabit noktalı hesapta bir hata oluşur ve bu hata,

$$-1.25025 - (-1.2421875) = -0.0080625 \text{ dir.}$$

Sonuç 16 bitlik bir bellekte saklanabilir fakat 8 bitlik bir alanda saklamamız gerekiyor. Sonucu 8 bite ölçeklememiz gerekiyor.

$$\alpha_{FxdPt} \times \beta_{FxdPt} = -10176_{10} (Q3.13) \gg 8 = \frac{-10176}{2^8} = -39_{10} (Q3.5) \text{ olur.}$$

Ölçeklenmiş sonuca  $\chi$  değeri de eklenir.

$$(\alpha_{FxdPt} \times \beta_{FxdPt}) + \chi_{FxdPt} = -39_{10} (Q3.5) + 83_{10} (Q3.5) = 44_{10} (Q3.5)$$

Sonuç:

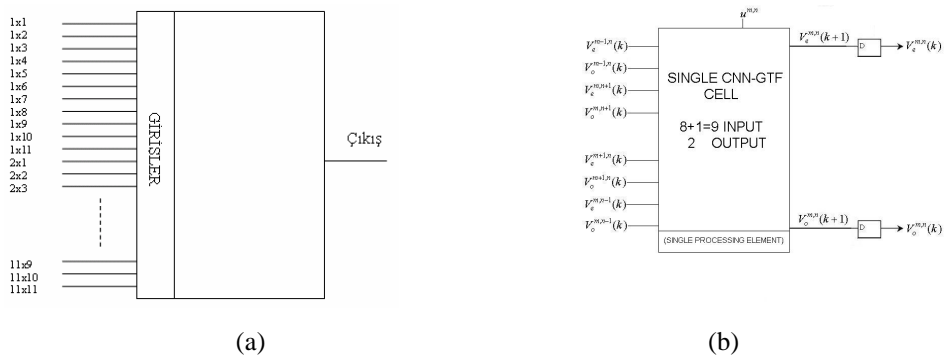
$$(\alpha_{FxdPt} \times \beta_{FxdPt}) + \chi_{FxdPt} = 44_{10} (Q3.5) = 1,375$$

Sabit noktalı işlem sonucunda oluşan hata ise;

$$((\alpha \times \beta) + \chi) - ((\alpha_{FxdPt} \times \beta_{FxdPt}) + \chi_{FxdPt}) = 1.34975 - 1.375 = -0.02525 \text{ olarak bulunur.}$$

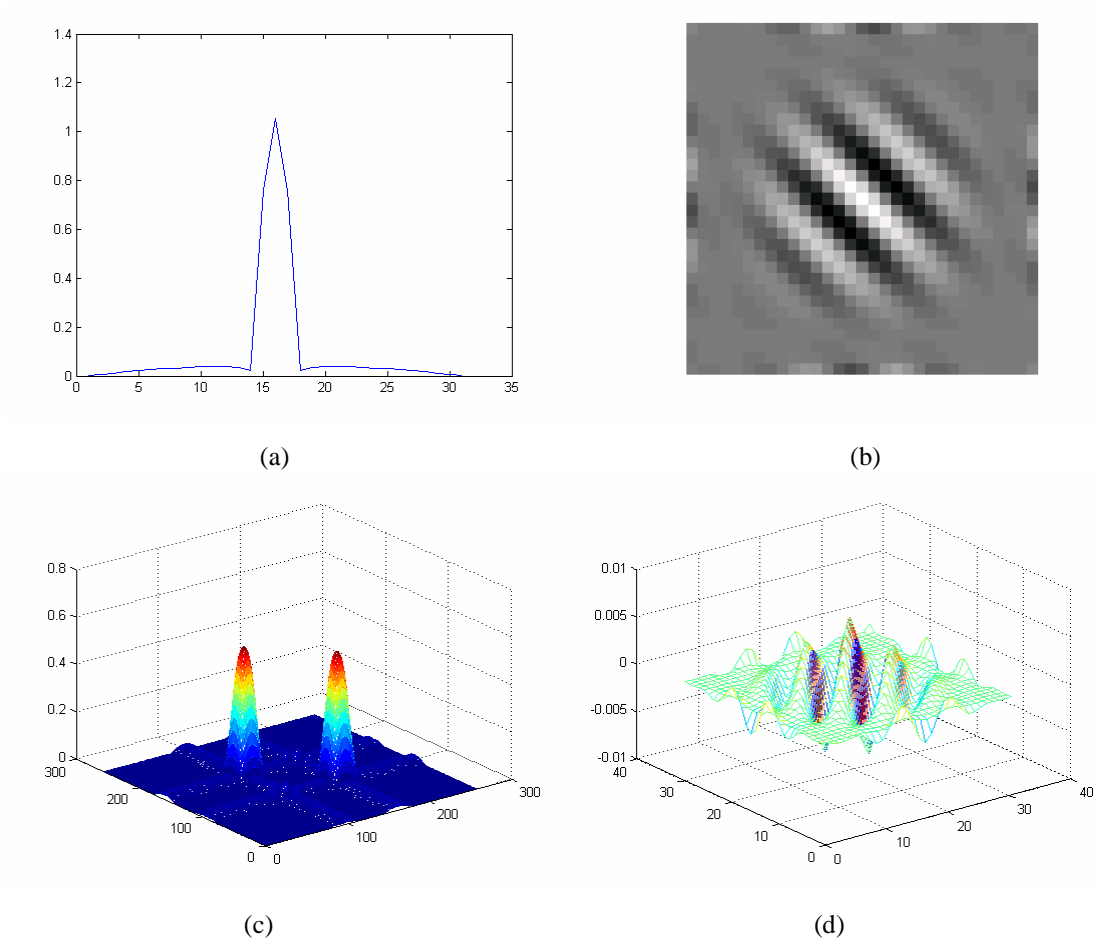
## 7. 2 BOYUTLU HSA GABOR SÜZGEÇİN MATLAB VE MODELSİM BENZETİMLERİ

4. bölümde gösterilen sayısal devre algoritmasının sonlu dürtü yanıtı görüntü işleme metotlarına olan üstünlüklerini şöyle gösterebiliriz. Gabor fonksiyonu üstel bir fonksiyon ile çarpıldığından karmaşık sayılar içermektedir. Bu nedenle sonlu dürtü yanıtı sayısal görüntü işleme yöntemi ile gerçekleştirilmesi oldukça güçtür. Klasik görüntü işleme ile gerçekleştirilmesi istendiği takdirde şablon boyutları belli bir boyutun üzerinde olmalıdır. Bu sınır 11x11 boyutudur. 11x11 boyutundaki bir şablonda 121 tane farklı değer ile görüntü konvolüsyon işlemini sokulması gerekmektedir. Bunun sonucunda da 256x256 boyutunda bir resim için  $11 \times 11 \times 256 \times 256 = 7929856$  tane çarpma işlemi ve  $10 \times 10 \times 256 \times 256 = 6553600$  tane toplama işlemi yapılması gerekmektedir. Bu çarpma ve toplama işlem sayıları günümüz sayısal teknolojisi için üstesinden gelinemez sayılardır fakat 121 girişli bir yapı hala büyük sorunlar doğurmaktadır (şekil 7.1).



Şekil 7.1: (a)121 Girişli FIR Gabor Yapısı , (b) HSA Gabor Tip Süzgeç birim işlem elemanı

121 girişli yapının içinde 121 tane katsayımız olmak zorunda. Gabor fonksiyonu daha büyük bir şablon ile gerçekleştirilmek istenirse bu sayılar artmaktadır. Aksine HSA Gabor Tip Süzgeç 5 katsayı ile gerçekleştirilebilmektedir.

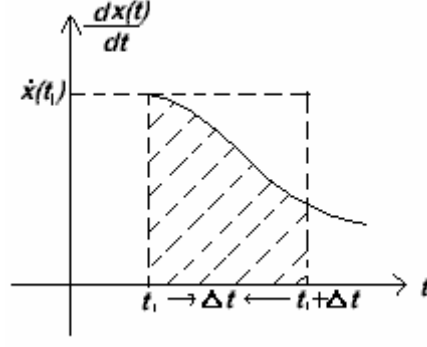


Şekil 7.2: (a) 31 uzunluklu bir sonlu süreli tek boyutlu alçak geçiren süzgecin frekans yanıtı, (b-d) (a)'da elde edilen süzgecin iki boyuta genişletilmiş durumunun dürtü yanıtı, (c) (b)'deki süzgecin frekans yanıtı.

## 7.1 MATLAB Benzetimi

### 7.1.1 HSA Benzetimi

(2.1) ile verilen standart HSA diferansiyel denkleminin sayısal çözümünün bulunabilmesi için yinelemeli yöntemler kullanılır. (2.1) standart HSA denkleminde sol taraf değeri olan  $\dot{x}_{ij}$ 'nin ayrık olarak çözülebilmesi için bir yaklaşıklık uygulanması gerekir. Yaklaşıklık olarak euler ileri yaklaşıklığı kullanılmıştır. Euler İleri Yaklaşıklığı; integralin dikdörtgen yaklaşıklığı ile hesaplanmasına karşı düşer.



Şekil 7.3. İntegralin dikdörtgen yaklaşıklığı ile hesaplanması

$$\int_{t_1}^{t_1+\Delta t} \dot{x}(t) dt \cong \Delta t \dot{x}(t_1)$$

$$\dot{x}(t) = \frac{dx(t)}{dt} \cong \frac{x(t + \Delta t) - x(t)}{\Delta t}$$

Diferansiyel haldeki durum denklemini fark denklemi haline getirmek için Eşitlik (2.1)' e Euler İleri Yaklaşıklığı'nı uygulanırsa Eşitlik 7.1 elde edilir.

$$\frac{dx_{ij}(t)}{dt} \cong \frac{x_{ij}(t + \Delta t) - x_{ij}(t)}{\Delta t} = -x_{ij}(t) + \sum_{k=-\sigma}^{\sigma} \sum_{l=-\sigma}^{\sigma} A(k, l) x_{i+k, j+l}(t) + \sum_{k=-\sigma}^{\sigma} \sum_{l=-\sigma}^{\sigma} B(k, l) u_{i+k, j+l} + z_{ij} \quad (7.1)$$

Denklem zamanda ayırık hale getirilirse eşitlik 7.2 elde edilir.

$$\dot{x}_{ij}(nT_s) \equiv \dot{x}_{ij}(t) \Big|_{t=nT_s}, \quad \Delta t \equiv T_s$$

$$\frac{x_{ij}((n+1)T_s) - x_{ij}(n)}{T_s} = -x_{ij}(nT_s) + \sum_{k=-\sigma}^{\sigma} \sum_{l=-\sigma}^{\sigma} A(k, l) x_{i+k, j+l}(nT_s) + \sum_{k=-\sigma}^{\sigma} \sum_{l=-\sigma}^{\sigma} B(k, l) u_{i+k, j+l} + z_{ij}$$

$$x_{ij}((n+1)T_s) = (1-T_s)x_{ij}(nT_s) + T_s \left( \sum_{k=-\sigma}^{\sigma} \sum_{l=-\sigma}^{\sigma} A(k,l)x_{i+k,j+l}(nT_s) + \sum_{k=-\sigma}^{\sigma} \sum_{l=-\sigma}^{\sigma} B(k,l)u_{i+k,j+l} + z_{ij} \right) \quad (7.2)$$

(7.2) fark denklemi ile sayısal olarak HSA standart denkleminin çözümü bulunur. Yinelemenin ne kadar yapılacağı benzetimin süresini belirlemektedir. Optimum zaman adımı [21] 'de verilmiştir.

(7.2) fark denkleminde  $x_{ij}$  durum değişkeninin ilk değer olarak resmin kendisi ya da tamamen sıfır alınır. Fark denkleminin sonucu, durum değişkeninin bir önceki değeri ile bir sonraki değeri arasında fark göz ardı edilecek seviyelere düştüğünde bulunmuş olunur. Sonuca ilk durumların katkısı yoktur sadece girişin katkısı kalmıştır.

Standart HSA diferansiyel denkleminin sol taraf değeri  $\dot{x}_{ij}$  zamanda türevi göstermektedir. Yinelemeli çözüm sonucunda  $\dot{x}_{ij}$  nin bir önceki değeri ile bir sonraki değeri arasında çok küçük fark olduğu için türev değeri sıfır olacaktır. (2.1) denklemi  $t \rightarrow \infty$  için sol tarafı sıfıra eşit olmaktadır.

$$0 = -x_{ij} + \sum_{C(k,l) \in S_r(i,j)} A(i,j;k,l)y_{kl} + \sum_{C(k,l) \in S_r(i,j)} B(i,j;k,l)u_{kl} + z_{ij} \quad (7.3)$$

$$x_{ij} = \sum_{C(k,l) \in S_r(i,j)} A(i,j;k,l)y_{kl} + \sum_{C(k,l) \in S_r(i,j)} B(i,j;k,l)u_{kl} + z_{ij}$$

$$\dot{x}_{ij} = -x_{ij} + \sum_{k=-\sigma}^{\sigma} \sum_{l=-\sigma}^{\sigma} a_{k,l}x_{i+k,j+l} + \sum_{k=-\sigma}^{\sigma} \sum_{l=-\sigma}^{\sigma} b_{k,l}u_{i+k,j+l} \quad (7.4)$$

(7.4) denkleminin ayrık Fourier dönüşümü (7.5) ile verilmiştir.

$$\frac{d}{dt} \tilde{\mathbf{X}}_t(\omega_x, \omega_y) = \tilde{\mathbf{A}}(\omega_x, \omega_y) \tilde{\mathbf{X}}_t(\omega_x, \omega_y) + \tilde{\mathbf{B}}(\omega_x, \omega_y) \tilde{\mathbf{U}}_t(\omega_x, \omega_y) \quad (7.5)$$

Zamana bağılı olmayan durağan  $\tilde{\mathbf{U}}_t(\omega_x, \omega_y) = \tilde{\mathbf{U}}(\omega_x, \omega_y)$  görüntüsü için (7.5) ifadesinin frekans domenindeki çözümü;

$\tilde{\mathbf{A}}(\omega_x, \omega_y) \neq 0$  için,

$$\tilde{\mathbf{X}}_t(\omega_x, \omega_y) = e^{\tilde{\mathbf{A}}(\omega_x, \omega_y)t} \tilde{\mathbf{X}}_0(\omega_x, \omega_y) + \frac{1}{\tilde{\mathbf{A}}(\omega_x, \omega_y)} \left[ e^{\tilde{\mathbf{A}}(\omega_x, \omega_y)t} - 1 \right] \tilde{\mathbf{B}}(\omega_x, \omega_y) \tilde{\mathbf{U}}(\omega_x, \omega_y) \quad (7.6)$$

$\tilde{\mathbf{A}}(\omega_x, \omega_y) = 0$  için,

$$\tilde{\mathbf{X}}_t(\omega_x, \omega_y) = \tilde{\mathbf{X}}_0(\omega_x, \omega_y) + t \underbrace{\tilde{\mathbf{B}}(\omega_x, \omega_y)}_{\text{FIR}} \tilde{\mathbf{U}}_t(\omega_x, \omega_y) \quad (7.7)$$

$\tilde{\mathbf{X}}_0(\omega_x, \omega_y)$  t=0 zamanındaki frekans düzlemindeki ilk koşul.

$\tilde{\mathbf{A}}(\omega_x, \omega_y) < 0$  için,

$e^{\tilde{\mathbf{A}}(\omega_x, \omega_y)t} \rightarrow 0$  olduğunda,

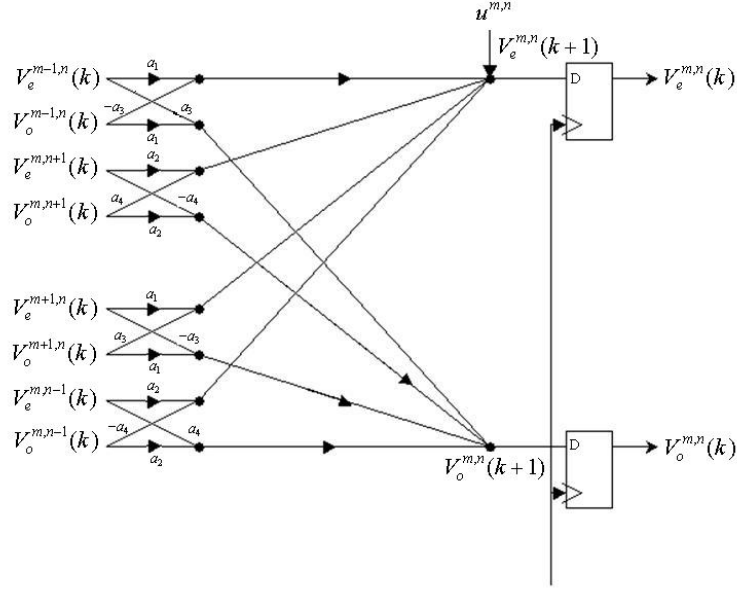
$$\tilde{\mathbf{X}}_\infty(\omega_x, \omega_y) \square \lim_{t \rightarrow \infty} \tilde{\mathbf{X}}_t(\omega_x, \omega_y) = - \left[ \frac{\tilde{\mathbf{B}}(\omega_x, \omega_y)}{\tilde{\mathbf{A}}(\omega_x, \omega_y)} \right] \tilde{\mathbf{U}}(\omega_x, \omega_y) \quad (7.8)$$

$- \left[ \frac{\tilde{\mathbf{B}}(\omega_x, \omega_y)}{\tilde{\mathbf{A}}(\omega_x, \omega_y)} \right] = \mathbf{H}(\omega_x, \omega_y)$  şeklinde düşünülürse  $\tilde{\mathbf{X}}_\infty(\omega_x, \omega_y) = \mathbf{H}(\omega_x, \omega_y) \tilde{\mathbf{U}}(\omega_x, \omega_y)$  olur.

$t \rightarrow \infty$  için H ile U nun uzamsal konvolüsyonu olur. Bu işlemde FIR süzgece karşılık düşer.

### 7.1.2 Kayan Noktalı Sayılar İle Benzetim

4. bölümde sayısal devre gerçekleştirme algoritması verilen iki boyutlu HSA Gabor süzgecin MATLAB ortamında benzetimi yapılacaktır.



Şekil 7.4: İki-boyutlu HSA Gabor Süzgecin Sayısal devre gerçekleştirilmesi

Katsayı değerleri bölüm 4 de hesaplandığı gibi şöyledir.

$$a_1 = \frac{\cos \omega_{x0}}{4 + \lambda^2}, \quad a_2 = \frac{\cos \omega_{y0}}{4 + \lambda^2}, \quad a_3 = \frac{\sin \omega_{x0}}{4 + \lambda^2}, \quad a_4 = \frac{\sin \omega_{y0}}{4 + \lambda^2}, \quad b = \frac{\lambda^2}{4 + \lambda^2}$$

$\omega_{x0}, \omega_{y0}$  : süzgecin geçirme frekanslarını,

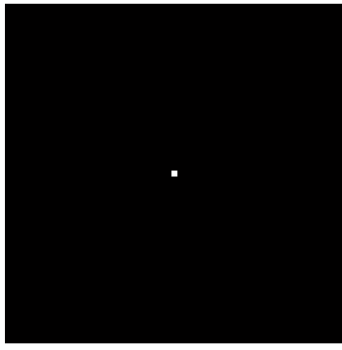
$\lambda$  : süzgecin bant genişliğini gösterir.

Yukarıdaki yapının donanım gerçekleştirilmesinden önce bilgisayar ortamında bir benzetimi yapılmıştır. Benzetim adımları şöyledir.

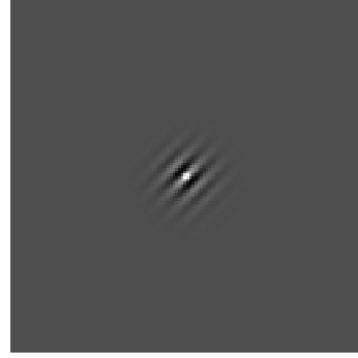
- Kullanılacak katsayılar hesaplanması
- Ara değerler için geçici durumların hazırlanması, ilk durum değerleri olarak tamamen sıfırlardan oluşmuş durumlar alınmıştır.
- Giriş resmi olarak merkezinde tek bir '1' olan bir resim kullanılmıştır. Bu resmin kullanılma nedeni bu süzgecin dürtü yanıtının hesaplanmasıdır.
- Yukarıdaki algoritma yapısı oluşturulmuştur iki iç içe döngü ile.
- 7.1.1 de gösterildiği gibi HSA diferansiyel denklemin çözümünün bulunabilmesi için gereken zaman ardı ardına dönen döngüler ile benzetimi yapılmıştır.
- Son olarak, elde edilen durumların görüntüleri ve frekans yanıtları çizdirilmiştir.



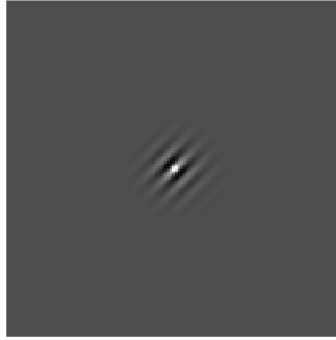
MATLAB ile yapılan bu benzetimin sonuçları aşağıda verilmiştir.



(a)

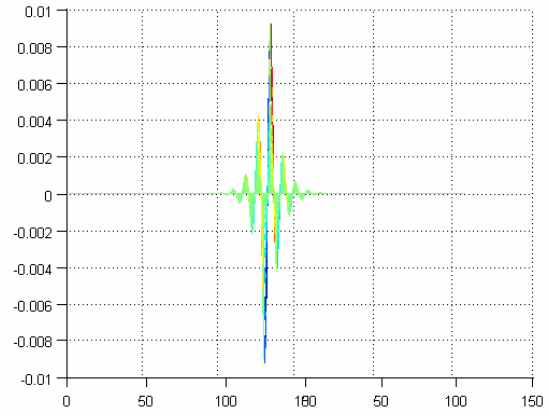
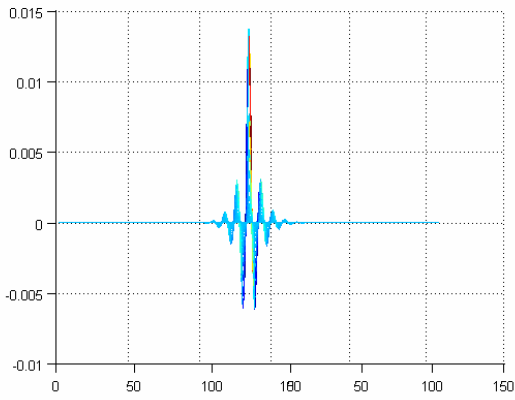


(b)

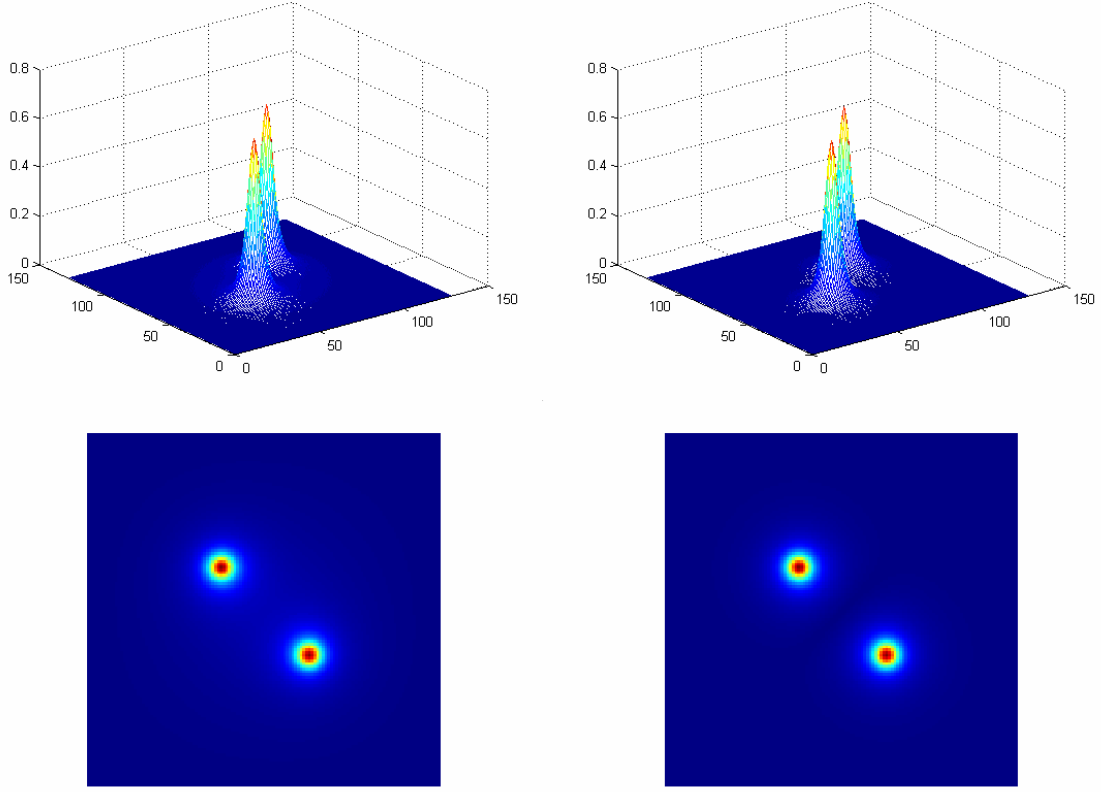


(c)

Şekil 7.5: (a) benzetim için kullanılan 128x128 boyutundaki giriş resmi. (b) ve (c) sırası ile çift ve tek Gabor' un görüntüleri



Şekil 7.6: Sırası ile çift ve tek Gabor' un dürtü yanıtları.



Şekil 7.7: Sırası ile çift ve tek gaborun frekans yanıtları.

Bu benzetimde kullanılan açı, bant genişliği ve katsayı değerleri şöyledir.

$$\text{Açı değerleri} \quad : \quad \omega_{x0} = \frac{\pi}{4}, \omega_{y0} = \frac{\pi}{4},$$

$$\text{Bant genişliği} \quad : \quad \lambda = 0.1,$$

$$\text{Katsayı değerleri} \quad : \quad a_1 = 0.1763, a_2 = 0.1763, a_3 = 0.1763, a_4 = 0.1763, b = 0.0025.$$

### 7.1.3 Sabit Noktalı Aritmetik ile Benzetim

Sabit noktalı benzetim, devre benzetiminin doğruluğu için yapılmıştır. MATLAB 7 programının Fixed Point Toolbox'ı kullanılmıştır. 7.1.2 deki algoritma (1.7) sabit nokta sayı formatında çalıştırılmıştır.

- Kullanılacak katsayılar double türden hesaplandıktan sonra “fi(d,1,8,7)” fonksiyonu ile 1.7 işaretli sabit noktalı sayıya dönüştürülür.
- Ara değerler için geçici durumların hazırlanması, ilk durum değerleri olarak tamamen sıfırlardan oluşmuş durumlar alınmıştır. Ara değer matrisleri “ fi ”

fonksiyonu ile 1.7 işaretli sabit noktalı sayıları saklayabilecek formata dönüştürülür.

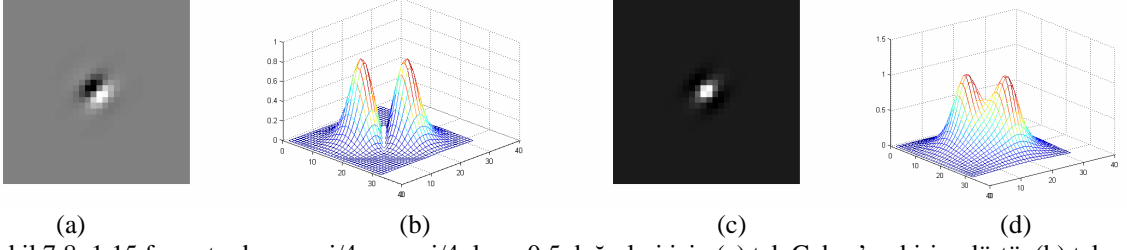
- Giriş resmi olarak merkezinde tek bir '1' olan bir resim kullanılmıştır. Bu resmin kullanılma nedeni bu süzgecin dürtü yanıtının hesaplanmasıdır. Giriş resmi "fi" fonksiyonu ile 1.7 işaretli sabit noktalı duruma dönüştürülmüştür.
- İç içe iki döngü ile algoritma yapısı oluşturulmuştur. Çarpım ve toplam sonuçları "fi" fonksiyonu ile 1.7 işaretli sabit noktalı hale dönüştürülmüştür.
- 7.1.1 de gösterildiği gibi HSA diferansiyel denklemin çözümünün bulunabilmesi için gereken zamanın benzetimi ardı ardına dönen döngüler ile yapılmıştır.
- Son olarak, elde edilen durumların görüntüleri ve frekans yanıtları çizdirilmiştir. Frekans yanıtlarının hesaplanabilmesi ( fft2 fonksiyonunun kullanılabilmesi ) için algoritma sonucunda elde edilen değerler double türe dönüştürülür.

1.7 sayı formatı ile ifade edilebilecek en küçük sayı 0.0078 ya da "00000001" dir. İfade edilebilecek en küçük sayının yeterince küçük olmaması nedeni ile 1.7 formatında bant genişliği  $\lambda$  'nın küçük değerler alamamasına neden olmaktadır.

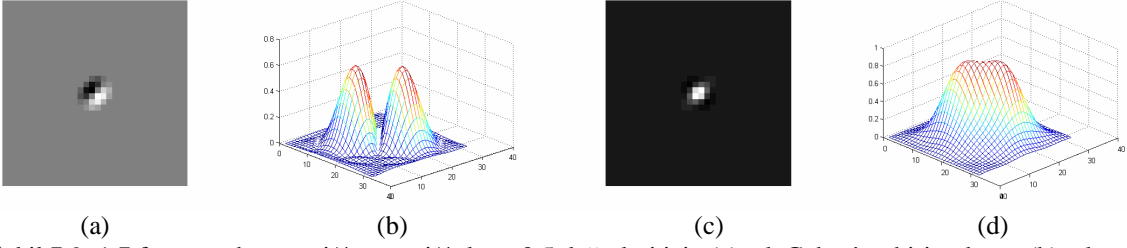
$\lambda$	$b = \lambda^2/4 + \lambda^2$		
	<b>double</b>	<b>1.7</b>	<b>1.15</b>
0.5	0.0588	0.0625	0.0588
0.4	0.0385	0.0391	0.0385
0.3	0.0220	0.0234	0.0220
0.2	0.0099	0.0078	0.0099
0.1	0.0025	0	0.0025
0.01	0.00003052	0	0.00002499
0.001	2.500e-007	0	0

Tablo 7.1:  $\lambda$  değerleri için elde edilen b değerleri

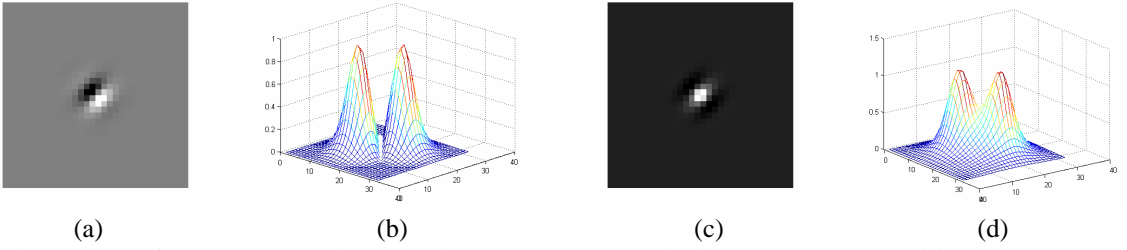
Şekil 7.8 ve 7.9 'da 1.7 ve 1.15 formatında benzetimi yapılmış HSA Gabor Tip Süzgeçlerin birim dürtü ve frekans yanıtları verilmiştir.



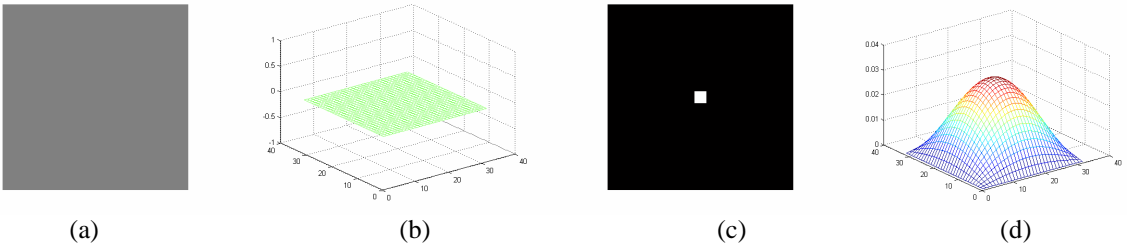
Şekil 7.8: 1.15 formatında  $w_x=\pi/4$ ,  $w_y=\pi/4$ ,  $\lambda=0.5$  değerleri için (a) tek Gabor'un birim dürtü, (b) tek Gabor'un frekans, (c) çift Gabor'un birim dürtü, (d) çift Gabor'un frekans yanıtları.



Şekil 7.9: 1.7 formatında  $w_x=\pi/4$ ,  $w_y=\pi/4$ ,  $\lambda=0.5$  değerleri için (a) tek Gabor'un birim dürtü, (b) tek Gabor'un frekans, (c) çift Gabor'un birim dürtü, (d) çift Gabor'un frekans yanıtları.



Şekil 7.10: 1.15 formatında  $w_x=\pi/4$ ,  $w_y=\pi/4$ ,  $\lambda=0.2$  değerleri için (a) tek Gabor'un birim dürtü, (b) tek Gabor'un frekans, (c) çift Gabor'un birim dürtü, (d) çift Gabor'un frekans yanıtları.

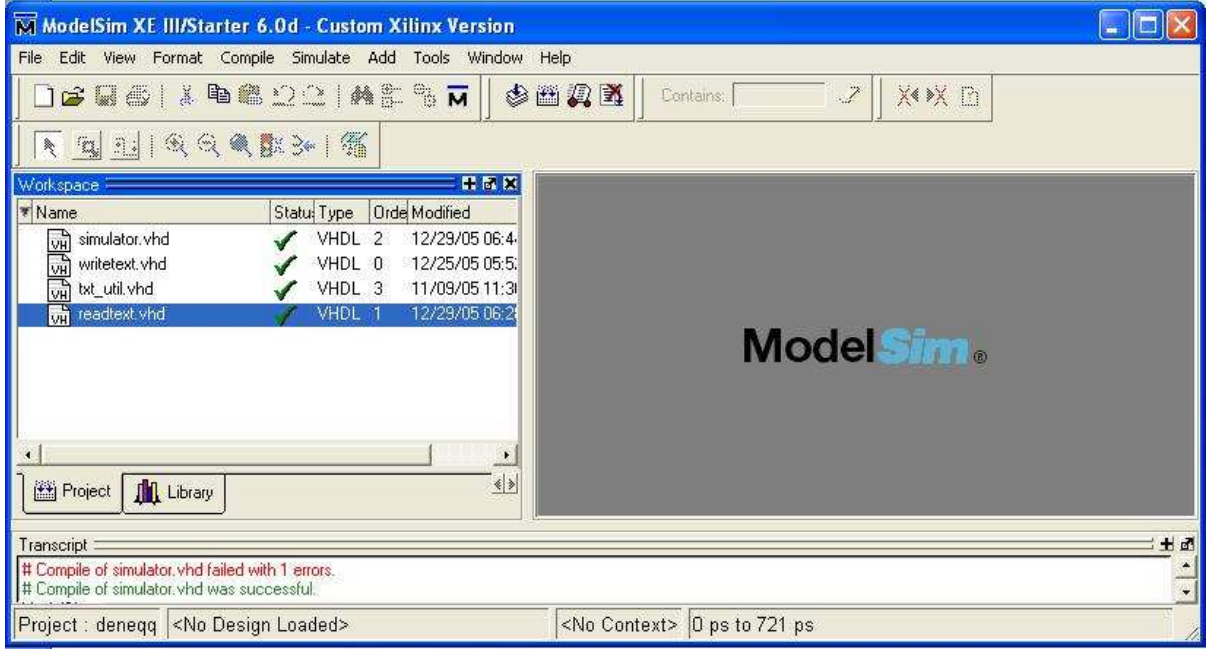


Şekil 7.11: 1.7 formatında  $w_x=\pi/4$ ,  $w_y=\pi/4$ ,  $\lambda=0.2$  değerleri için (a) tek Gabor'un birim dürtü, (b) tek Gabor'un frekans, (c) çift Gabor'un birim dürtü, (d) çift Gabor'un frekans yanıtları.

Şekil 7.11'de 1.7 formatının minimum ve ifade aralığı sorunu görülmektedir.  $\lambda = 0.2$  için 1.7 formatındaki  $b$  değeri 0.0078 dir. "0.0078" değeri 1.7 formatındaki minimum artım miktarıdır. Şekil 7.4 de verilen algoritmaya değerler uygulandığında  $V_0$  ve  $V_e$  değerleri sabit kalmaktadır. Sabit kalmanın nedeni ise artım minimum artım miktarını aşacak bir değer üretilmemesidir.

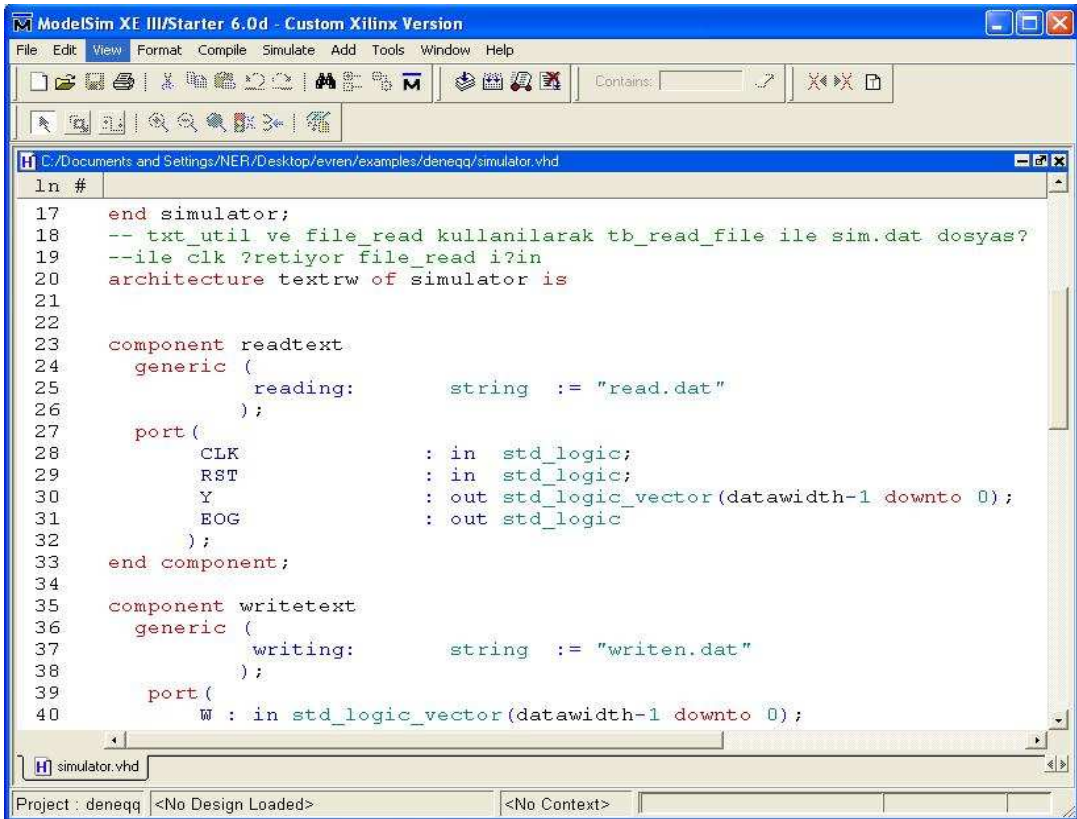
## 7.2 Modelsim Benzetimi

Modelsim Mentor Graphics Şirketi tarafından tamamen FPGA benzetimi için geliştirilmiş bir benzetim aracıdır. Modelsim, FPGA üzerinde gerçekleştirme yapmadan önce yazılan VHDL ve Verilog kodlarının benzetimini tıpkı donanım üzerindeki gibi yapar. Modelsimde yapılan benzetime geçmeden önce Modelsimin kısa bir tanıtımı yapılmıştır.



Şekil 7.12: Modelsim giriş sayfası

Bu sayfa Modelsim ilk açıldığında görünmektedir. Şu anda sayfanın solundaki alanda benzetimini yaptığım basit bir giriş işaretinin sabit bir sayı ile çarpılıp çıkış işareti oluşturulmaktadır. Giriş işareti metin dosyası halinde tutulmaktadır. Benzetim sırasında metin dosyası bir işaret kaynağı görevi görmektedir. Çıkış ise başka bir metin dosyasına yazılmaktadır.



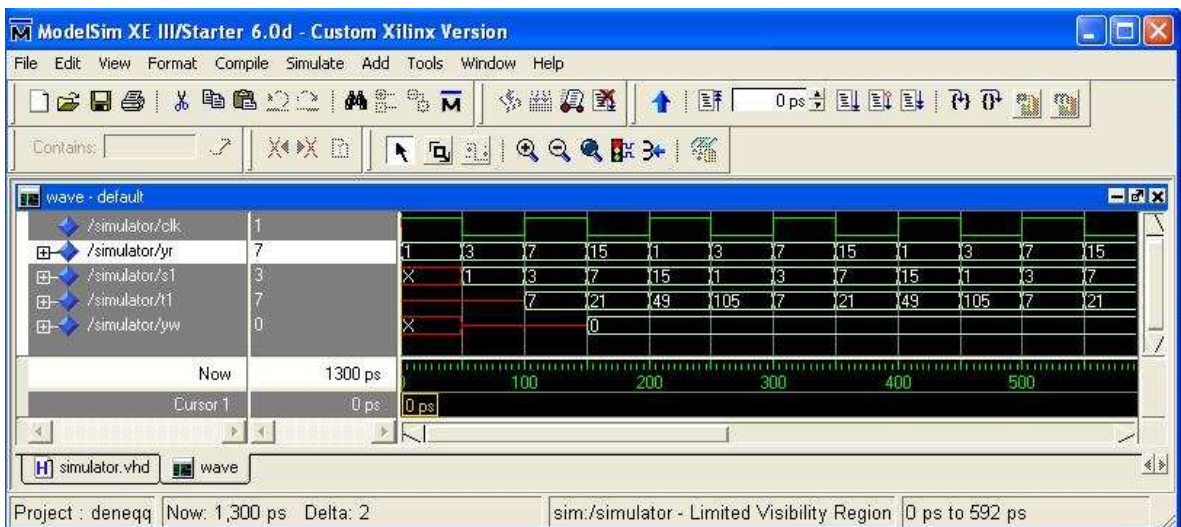
```

ModelSim XE III/Starter 6.0d - Custom Xilinx Version
File Edit View Format Compile Simulate Add Tools Window Help
C:/Documents and Settings/NER/Desktop/evren/examples/deneqq/simulator.vhd
ln #
17 end simulator;
18 -- txt_util ve file_read kullanılarak tb_read_file ile sim.dat dosyas?
19 --ile Clk ?retiyor file_read i?in
20 architecture textrw of simulator is
21
22
23 component readtext
24   generic (
25     reading:      string := "read.dat"
26   );
27   port (
28     CLK           : in  std_logic;
29     RST           : in  std_logic;
30     Y             : out std_logic_vector (datawidth-1 downto 0);
31     EOG          : out std_logic
32   );
33 end component;
34
35 component writetext
36   generic (
37     writing:      string := "writen.dat"
38   );
39   port (
40     W : in std_logic_vector (datawidth-1 downto 0);

```

Şekil 7.13: Modelsim içinde yazılmış VHDL kodu.

Yazılan VHDL kodları önce derlenmekte ve yazım hatalarına taranmaktadır. Herhangi bir hatada derleyici uyarıyı şekil 7.12 de ki resmin alt kısmında görüldüğü gibi vermektedir. Herhangi bir uyarı olmadığı zaman derlenme işlemi başarılıymış sayılmaktadır. Derleme işlemi başarılı olduktan sonra sıra benzetim işlemine gelmektedir. Benzetim işlemi için ayrı bir pencere açılır ve bu pencerede tanımlanmış olan bütün işaretler gözükmektedir.

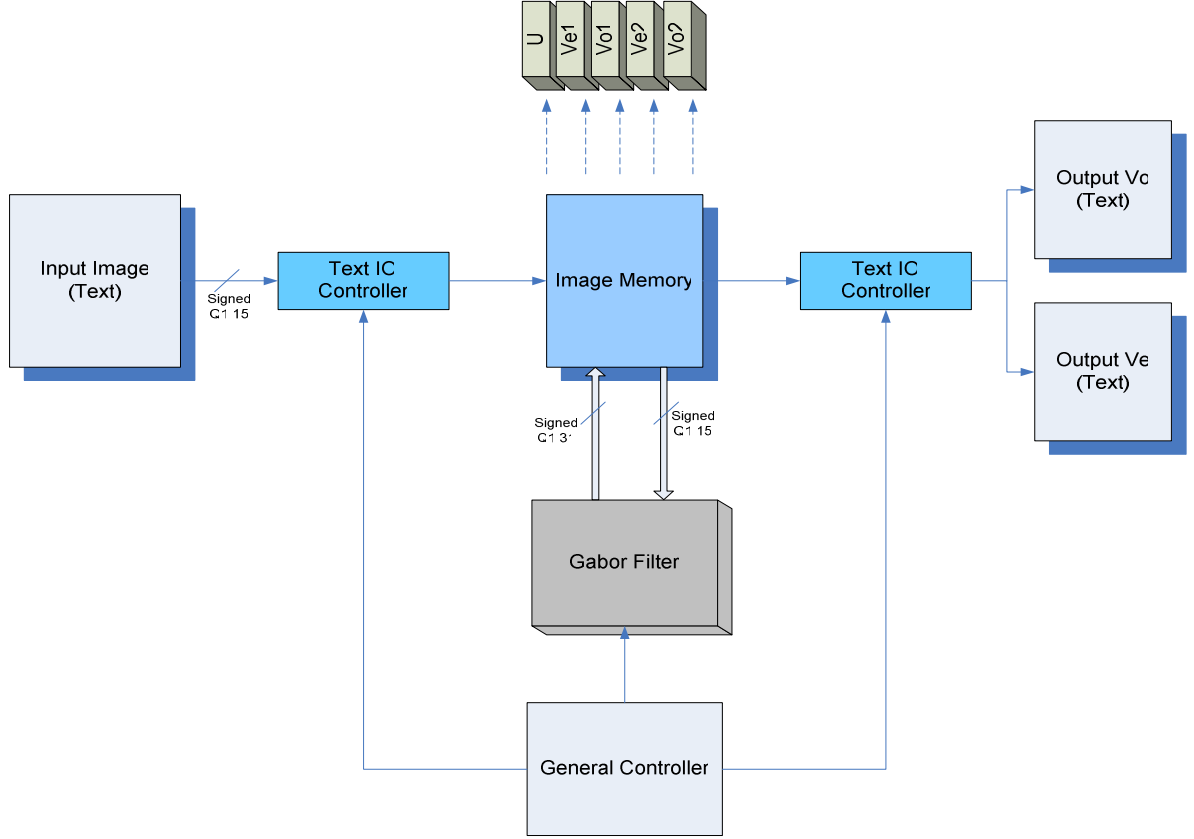


Şekil 7.14: Modelsim benzetim sonuçları.

Bu örnek benzetimde giriş işareti 7 ile çarpılıp sonuç çıktı olarak çıkış işaretine verilmektedir.

### 7.2.1 HSA Gabor Tip Süzgecin Modelsim Benzetimi

Modelsim de yapılacak olan benzetimin amacını daha önce tartışmıştık Bu benzetimde şu adımları izleyeceğiz. Modelsim ile yapılan benzetimin genel yapısı şekil 7.15 ' te görüldüğü gibidir.



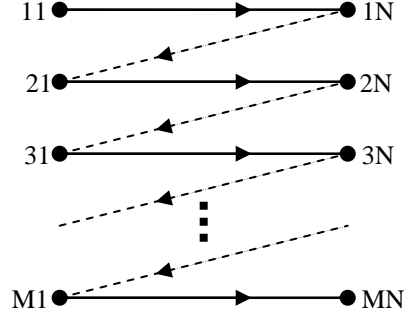
Şekil 7.15: Genel yapı

Benzetim işleminin genel yapısı MATLAB ile elde edilen giriş görüntüsü, metin dosya kontrolleri, genel kontrol, durum bellek alanları, Gabor filtre algoritması ve çıkış metin dosyalarından oluşmaktadır.

Giriş görüntüsü şekil 7.16 'teki tarama yolu ile elde edilmiştir. Her bir piksel işaretli sabit nokta 1.15 formatında ifade edilmektedir. Bu işlemler sonunda görüntü şekil 7.17 teki tek boyutlu dizi halini alır.

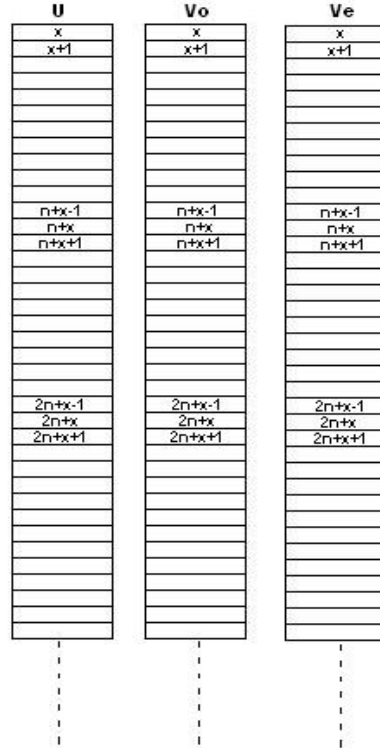
Metin dosya kontrolleri ile metin dosyadan okunan giriş sayısal görüntü piksel değerleri sayısal görüntü durum belleklerine yerleştirilir. Durum değişkenleri sırası ile giriş U, birinci Vo, ikinci Vo, birinci Ve ve ikinci Ve dir. Vo ve Ve iki katmanlı HSA yapısı ile hesaplanan

Gabor filtresinin gerçek ve sanal kısımlarıdır. İkişer tane olmasını nedeni ise belli bir iterasyonla hesaplama gerekliliğidir.



Şekil 7.16: Satır satır tarama işlemi.

Hesaplamalar sonunda oluşan  $V_o$  ve  $V_e$  değerleri metin dosya kontrolleri ile çıkış  $V_o$  ve  $V_e$  metin dosyalarına kayıt edilir. Çıkış metin dosyaları MATLAB ile ters işleme sokularak görüntüler elde edilir.

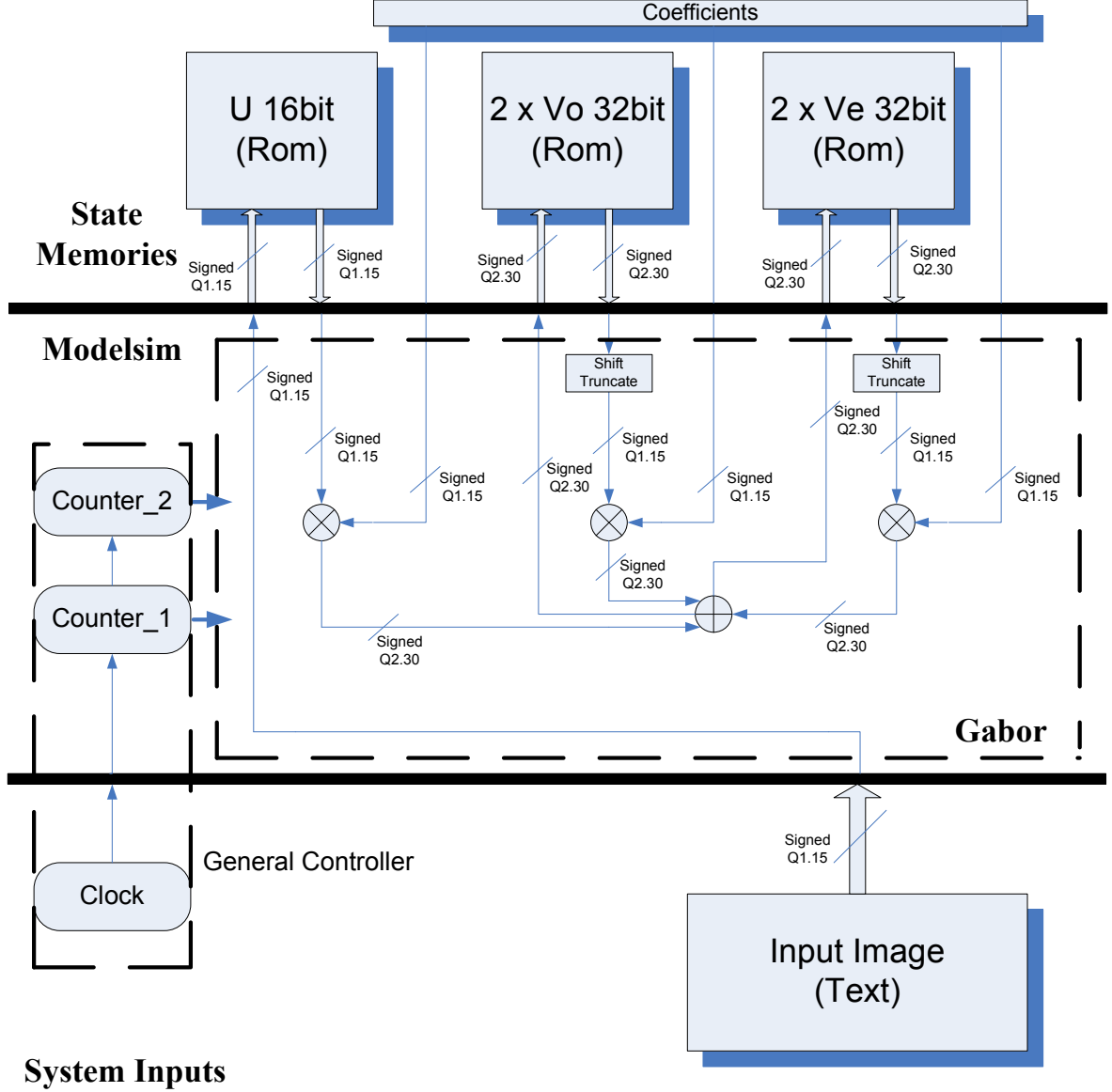


Şekil 7.17: Giriş ve durum kayıtçı tabloları.

Modelsim benzetim işaret akışları şekil 17 de verilmiştir. Giriş görüntüsü durum makinesinin ilk durumu için giriş U bellek bölgesine yazılır.  $V_{o1}$  ve  $V_{e1}$  ilk durum değerleri sıfır olarak

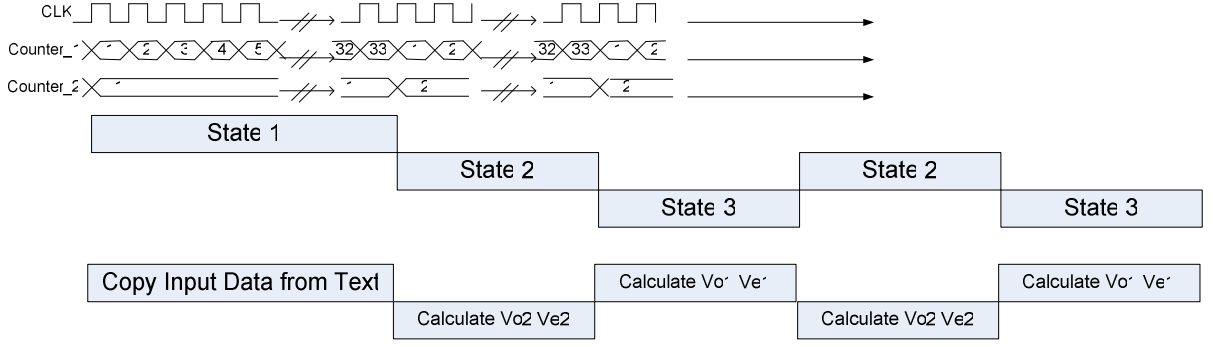


doldurulmuştur. Durum makinesinin ikinci durumu için U, Vo1, Ve1 ve katsayılar ile Vo2 ve Ve2 durum değişkenleri bellek bölgeleri doldurulmaktadır. Durum makinesinin üçüncü durumunda U, Ve2, Vo2 ve katsayılar ile Vo1 ve Ve1 in yeni değerleri hesaplanmaktadır.



Şekil 7.18: İşaret akışı

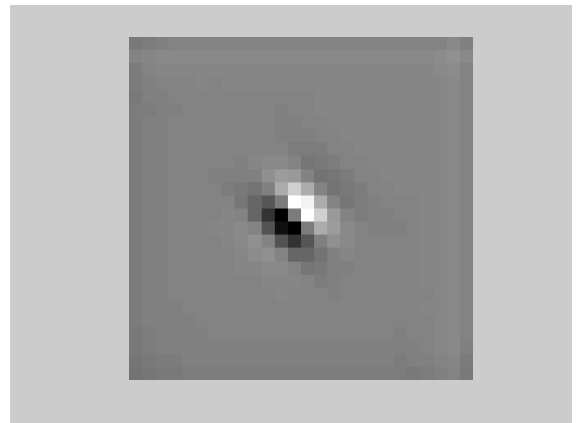
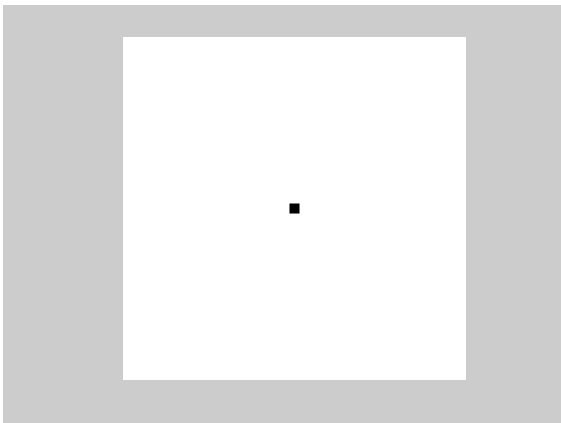
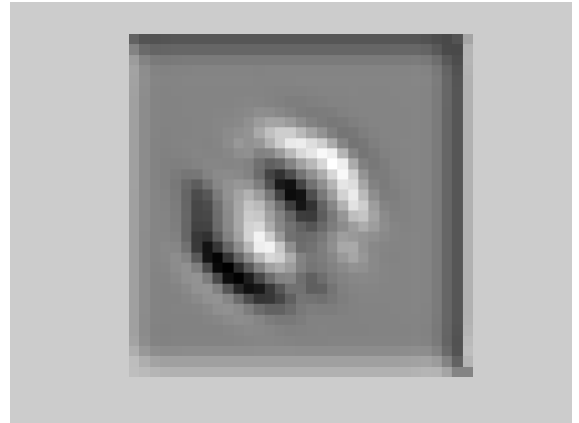
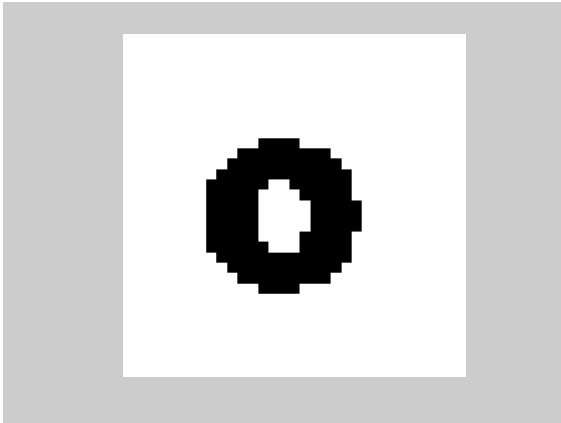
Katsayılar ile piksel değerleri çarpım sonuçları Q2.30 formatında olur. Bu değerlerin yeni durumda tekrar Q1.15 formatı olarak işleme sokulması için ötelenip kesilmesi gerekmektedir. Durum makinesinin durumları arasındaki geçişi "Counter\_2" yapmaktadır.

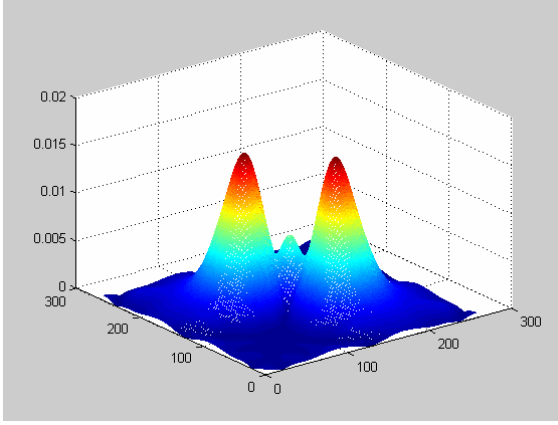


Şekil 7.19: İşaret durumları

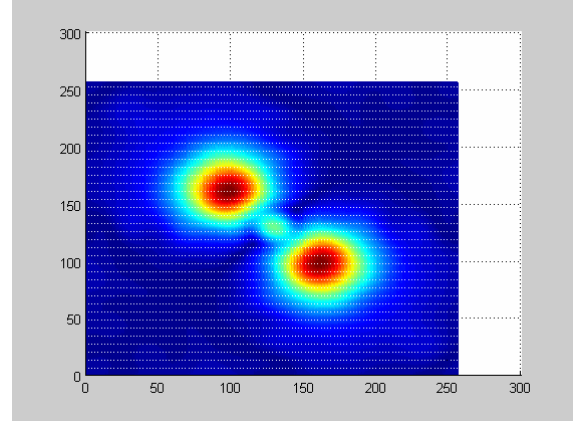
İşaret ve veri akışı şekil 18 te gösterilmiştir. Her saat darbesini yükselen kenarında, Counter\_1 artmaktadır. Counter\_1 taştıktan sonra Counter\_2 artmaktadır. Counter\_1 in her taşmasında yeni duruma geçilmektedir. İşlemlerin sonuçlanması için Counter\_2 nin taşması beklenmektedir. Counter\_1 ayrıca bellek dizilerinin adreslenmesi içinde kullanılmaktadır. Hsa Gabor tip Süzgecin sonucunun hesaplanması için gereken iterasyon sayısı Counter\_2 nin taşma değeridir.

Modelsim benzetimi sonrası elde edilen giriş ve çıkış görüntüleri.

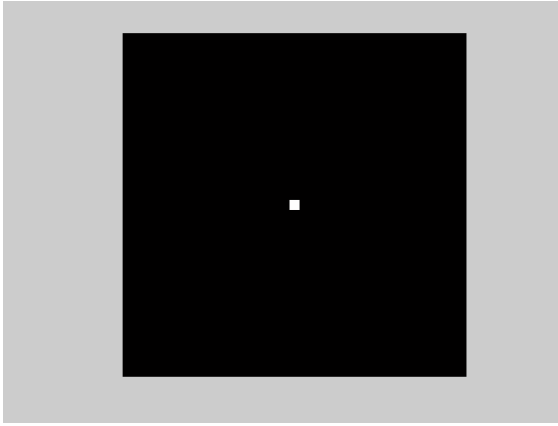




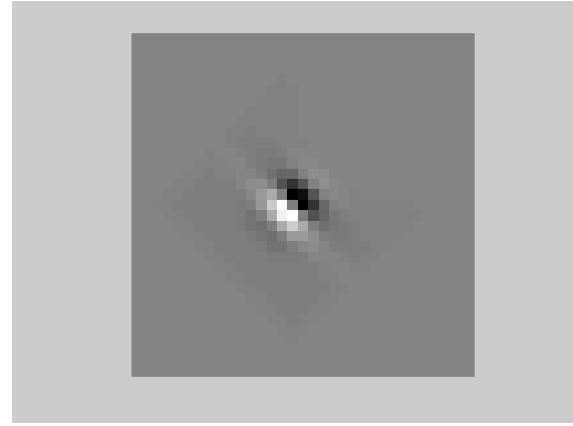
(e)



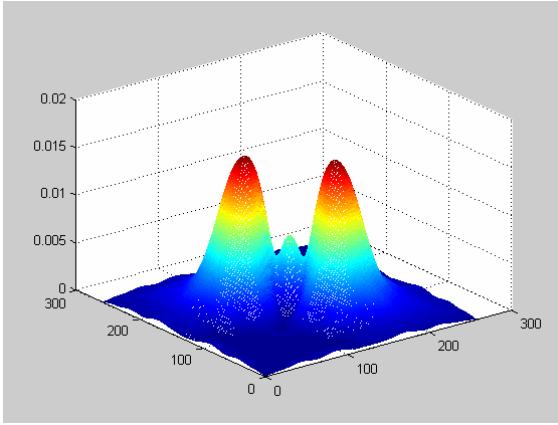
(f)



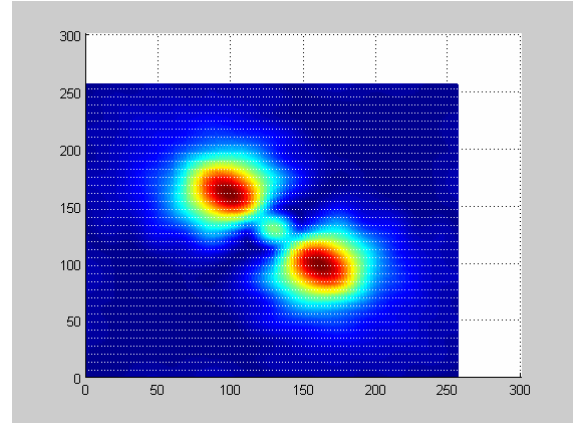
(f)



(h)

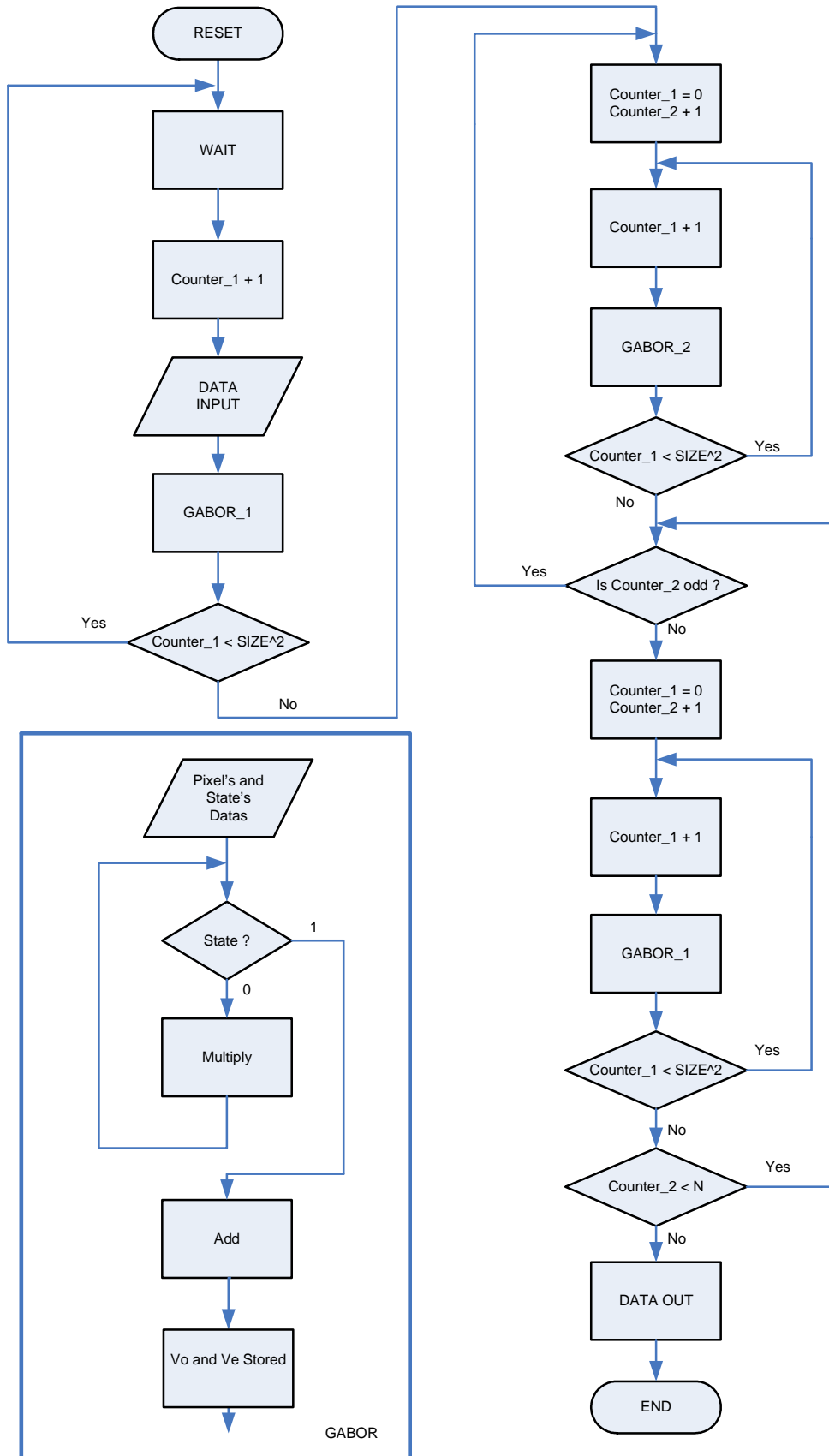


(i)



(j)

Şekil 7.20: Modelsim benzetim giriş ve çıkış sayısal görüntüleri. Açı değerleri x yönünde  $45^0$ , y yönünde ise  $-45^0$ , bant genişliği ise 0.1 alınmıştır.



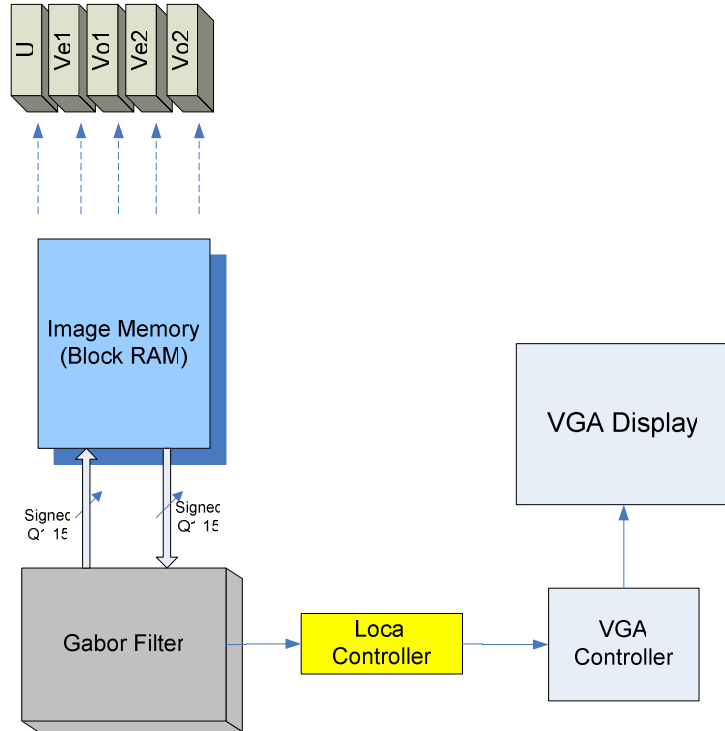
Şekil 7.21: Gabor Filtresi için kullanılan algoritma.

### 7.3. HSA Gabor Tip Süzgecin FPGA Benzetimi

Benzetim için DIGILENT firmasının üretmiş olduğu SPARTAN-3 Bordu kullanılmıştır.

Bordun özellikleri;

- Xilinx Spartan-3 XC3S200-FT256
  - On iki 18 Bit Çarpıcı
  - D Digital Clock Managers (DCM)
  - 216 Kbits Block Ram
  - 30 Kbits Distributed Ram
  - 500 MHz iç saat hızı
  - 200K Kapı
- 2 Mbit Platform Flash (XCF02S)
- 8 anahtar, 4 düğme, 9 LED, 4-hane yedi parçalı gösterge
- Seri port, VGA portu, PS/2 fare/klavye portu
- 1 Mbyte 10 ns SRAM (256Kb x 32)



Şekil 7.22: Gabor Filtresi FPGA benzetimi genel yapısı.

Kullanılan yazılım ISE 7.1 programıdır. ISE Xilinx firmasının geliřtirdiđi derleyici, sentezleyici, programlayıcı bulunmaktadır. VHDL, Verilog veya řematik ile yapılan tasarımlar bu program ile FPGA iine JTAG ara yüzü ile yüklenmektedir.

Yapılan işlemler;

- İşlenecek sayısal görüntü MATLAB yardımı ile řekil 7.16 teki gibi satır satır taranarak řekil 7.17 daki tek boyutlu dizi elde edilir.
- Dizinin elemanları işaretili Q1.15 formatına dönüřtürülür.
- Yazılan VHDL kodu iine elde edilen görüntü eklenir.
- Sentezlenen VHDL kodu ISE ile JTAG üzerinden borda yüklenir.
- Sonuç 3 bit 8 renk olarak VGA ekranda görüntülenir.

## 8. SONUÇLAR

HSA yapısı görüntü işleme uygulamaları için çok elverişlidir. Çünkü görüntü işleme uygulamaları HSA gibi görüntü pikselleri için bağımsızdır. Görüntüdeki her piksel HSA ağındaki bir düğüme eşlenebilir ve paralel analog hesaplama sayısal işlemcilerle göre büyük bir hız üstünlüğü sağlar. Klasik görüntü işlemedeki süzme işlemi sonlu dürtü yanıtı ile resmin konvolüsyonu yapılmaktadır. Bu yöntemde sınırlı boyutta şablon kullanılması zorunluluğu yüzünden istediğimiz bazı fonksiyonları elde etmekte güçlük çekilmektedir. HSA'nın analog yapısı, bir başka deyişle sonsuz uzunluklu dürtü yanıtı yapısı görüntü işlemede sonlu dürtü yanıtı şablonlarla gerçekleştiremeyeceğimiz fonksiyonları gerçekleştirme imkânı vermektedir. Sonlu süreli dürtü yanıtına sahip (FIR) yöntemleri ile gerçekleştirmekte zorluk çekeceğimiz fonksiyonlardan biri Gabor fonksiyonudur.

Gabor fonksiyonu alçak geçiren bir süzgecin üstel bir fonksiyon ile çarpılarak bant geçiren durumuna getirilmesi ile elde edilmektedir. Gabor frekans seçiciliği yanında görüntü içinde yön seçiciliği de yapmaktadır. Bu özellikleri yüzünden görüntü işleminin içinde birçok alanda kullanılmaktadır. Örnek olarak görsel korteks, hareket algılama, parmak izi tanıma, karakter tanıma gibi. Gabor fonksiyonu üstel bir fonksiyon ile çarpıldığından karmaşık sayılar içermektedir. Bu nedenle sonlu dürtü yanıtı görüntü işleme yöntemi ile gerçekleştirilmesi oldukça güçtür. Klasik görüntü işleme ile gerçekleştirilmek istendiği takdirde şablon boyutları belli bir boyutun üzerinde olmalıdır. Bu sınır 11x11 boyutudur. 11x11 boyutundaki bir şablonda 121 tane farklı değer ile görüntü konvolüsyon işlemi yapılması gerekmektedir. Bunun sonucunda da 256x256 boyutunda bir resim için  $11 \times 11 \times 256 \times 256 = 7,929,856$  tane çarpma işlemi ve  $10 \times 10 \times 256 \times 256 = 6,553,600$  tane toplama işlemi yapılması gerekmektedir. Bu çarpma ve toplama işlem sayıları günümüz sayısal teknolojisi için üstesinden gelinemez sayılardır fakat 121 girişli bir yapı hala büyük sorunlar doğurmaktadır.

Açıklanan nedenlerden dolayı Gabor fonksiyonunu daha kullanışlı hale getirmek için B. Shi HSA Gabor Tip süzgeçleri geliştirmiştir. İki katmanlı HSA yapısı kullanılarak geliştirilen bu yapı aynı anda Gabor fonksiyonun hem gerçek hem de sanal kısmını hesaplayabilmektedir. Ayrıca yapı saf analog sistem olduğundan sonuca nano saniyeler mertebesinde ulaşabilmektedir.

Geliştirilen bu analog yapının, sayısal bir yapıya dönüştürülebilmektedir. HSA denklemlerinde türev ifadesi yerine yaklaşıklık metotları uygulanarak HSA analog sistemi sayısal olarak elde edilebilmektedir. 4. bölümde gösterilen sayısal gerçekleştirilmesi, sonlu dürtü yanıtı görüntü işleme metotlarına karşı üstünlüğü çok açık görülmektedir.

Tezin son bölümünde ise elde edilen HSA Gabor tip süzgecin sayısal devre üzerinde gerçekleştirilmesi yapılmıştır. Benzetim sonuçları yapılan işlemlerin doğru olduğunu göstermektedir.

Gelecek çalışmalar ise Gabor süzgecin uygulamaları yönünde olabilir. Örnek olarak parmak izi tanımlama sistemi düşünülmektedir.



**KAYNAKLAR**

- [1] L. O. Chua and L. Yang, "Cellular Neural Networks: Theory", IEEE Trans. CAS, vol. 35, no. 10, pp. 1257-1272, Oct. 1988
- [2] M. Clark, A. C. Bovik and W.S. Geisler, "Texture segmentation using a class of narrowband filters", Proceedings of Int. Conf. Acoustics, Speech, Signal Processing, pp. 571-574, 1987.
- [3] A. K. Jain and S. K. Bhattacharjee, "Text segmentation using Gabor filters for automatic document processing", Machine Vision Applicat., vol. 5, no. 3, pp. 169-184, 1992.
- [4] M. Porat and Y. Y. Zeevi, "localized texture processing in vision: Analysis and synthesis in the Gaborian space", IEEE Transactions on Biomedical Engineering, vol. 36, no. 1, pp. 115-129, Jan. 1989.
- [5] R. Mehrotra, K. R. Namuduri and N. Ranganathan, "Gabor filter-based edge detection", Pattern Recognition, vol. 25, no. 12, pp. 1479-1494, 1992.
- [6] J. G. Daughman, "Complete discrete 2-D Gabor transforms by neural Networks for image analysis and compression", IEEE Trans. Acoust. Speech Signal Processing, vol. 36, no. 7, pp. 1169-1179, July 1988.
- [7] D. J. Heeger, "Model for the extraction of image flow", Journal of the Optical Society of America, vol. 4, no. 8, pp. 1455-1471, 1987.
- [8] D. P. Casasent, J. S. Smokelin and A. Ye, "Wavelet and Gabor transforms for detection", Opt. Eng., vol. 31, no. 9, pp. 1893-1898, 1992.
- [9] B. J. Super and A. C. Bovik, "Three-dimensional orientation from texture using Gabor wavelets", SPIE Conf. Visual Communications Image Processing, pp. 574-586, 1991.
- [10] S. Marceljia, "Mathematical description of the responses of simple cortical cells", J. Opt. Soc. Amer., vol. 70, pp. 1297-1300, Nov. 1980.
- [11] D. Gabor, "Theory of Communication", J. Inst. Elect. Eng. London, vol. 93, pp. 429-457, 1946.
- [12] J. G. Daugman, "Uncertainty Relation for Resolution in Space, Spatial Frequency and Orientation Optimized by Two-Dimensional Visual Cortical Filters", Journal of the Optical Society of America, vol. 2, no. 7, pp. 1160-1169, 1985.
- [13] L. O. Chua and L. Yang, "Cellular neural networks: Applications," IEEE Trans. Circuits Syst., vol. 35, pp. 1273-1290, Oct. 1988.

- [14] B. E. Shi, "Spatio-temporal filtering with cellular neural networks," Ph.D. dissertation, Dept. Elect. Eng. Computer Sci., Univ. California Berkeley, 1994.
- [15] K. R. Crouse and L. O. Chua, "Methods for image processing and pattern formation in cellular neural networks: A tutorial," *IEEE Trans.Circuits Syst. I*, vol. 42, pp. 583–601, Oct. 1995.
- [16] B. E. Shi, "Gabor-type filtering in space and time with cellular neural networks", *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, vol. 45, no. 2, pp. 121-132, Feb. 1998.
- [17] K. R. Crouse and L. O. Chua, "Methods for image processing and pattern formation in cellular neural Networks: A tutorial" , *IEEE Transactions on Circuits and Systems-I: Fundamental theory and Applications*, vol. 42, pp. 583-601, Oct. 1995
- [18] A.D. Poularikas, "The Hankel Transforms", in *The Handbook of Formulas and Tables for Signal Processing*, Ed. Alexander D. Poularikas, CRC Pres, Boca Raton, 199.
- [19] B. E. Shi, "Gabor-type image filtering with cellular neural Networks", *IEEE International Symposium on Circuits and Systems Proceedings (ISCAS'96)*, vol. 3, pp. 558-561, May 1996.
- [20] P. Arato, T. Visegrady, I. Jankovits, Sz. Szigeti, "High-Level Aynthesis of Pipelined Datapaths", Edited by P. Arato, Panem Budapest, 2000
- [21] E. Saatci and V. Tavsanoğlu, "On The Optimal Choice of Integration Time-step for Raster Smilation of a CNN for Gray Level Image Processing", *IEEE International Symposium on Circuit and Systems (ISCAS'02)*, pp. 625-628, Arizona, USA, 26-29 May 2002.

## İnternet

[www.vhdl-online.de/tutorial/](http://www.vhdl-online.de/tutorial/)

[www.xilinx.com](http://www.xilinx.com)

[www.digilentinc.com](http://www.digilentinc.com)

[www.model.com](http://www.model.com)

<http://www.superkits.net/whitepapers/Fixed%20Point%20Representation%20&%20Fractional%20Math.pdf>

**ÖZGEÇMİŞ**

Ad Soyad	Evren CESUR
Doğum Tarihi	12.09.1980
Doğum Yeri	Fatih – İstanbul
Lisans	Sakarya Üniversitesi (1997 - 2002) Elektrik- Elektronik Mühendisliği
İş Deneyimi	Araştırma Görevlisi (2005 - Devam) Yıldız Teknik Üniversitesi Elektronik ve Haberleşme Bölümü Devreler ve Sistemler A.B.D