

# Bölüm 10 FBs- PLC Yüksek-Hızlı Sayıcı ve Zamanlayıcı

## 10.1 FBs-PLC Yüksek-Hızlı Sayıcı

Sıradan bir PLC'nin yazılım sayacının frekansı sadece Hz'nin on katlarına ulaşabilir (tarama zamanına bağlı olarak). Eğer giriş sinyalinin frekansı bundan yüksek ise, yüksek hızlı sayıcı (HSC) kullanmak gerekir, aksi takdirde saymada kayıplar meydana gelebilir. PLC için genelde iki tip HSC vardır. Yüksek hızlı donanım sayıcı HHSC; (özel donanım devresi) ve yüksek hızlı yazılım sayıcı SHSC (sinyal konum değiştirdiğinde arttırma/azaltma sayım işlemini yürütmek için CPU'yu kesen) kullanılır. Fbs PLC 4 HHSC tipi sayıcı, 4 SHSC tipi sayıcıyı destekler. Hepsisi 32 bit high speed sayıcılarıdır.

### 10.1.1 FBs- PLC Yüksek-Hızlı Sayıcının Sayım Modları

Aşağıdaki tabloda gösterildiği gibi, HHSC ve SHSC lerin toplam 8 tip sayma modu vardır, tabloda sayım modları açıklanmıştır;

Counting Mode			HHSC (HSC0~HSC3)	SHSC (HSC4~HSC7)	Counting Waveform	
					Up Counting (+1)	Down Counting (-1)
Up-down pulse	MD 0	U/D	○	○		
	MD 1	U/Dx2	○			
Pulse-direction	MD 2	P/R	○	○		
	MD 3	P/Rx2	○			
AB phase	MD 4	A/B	○	○		
	MD 5	A/Bx2	○			
	MD 6	A/Bx3	○			
	MD 7	A/Bx4	○			

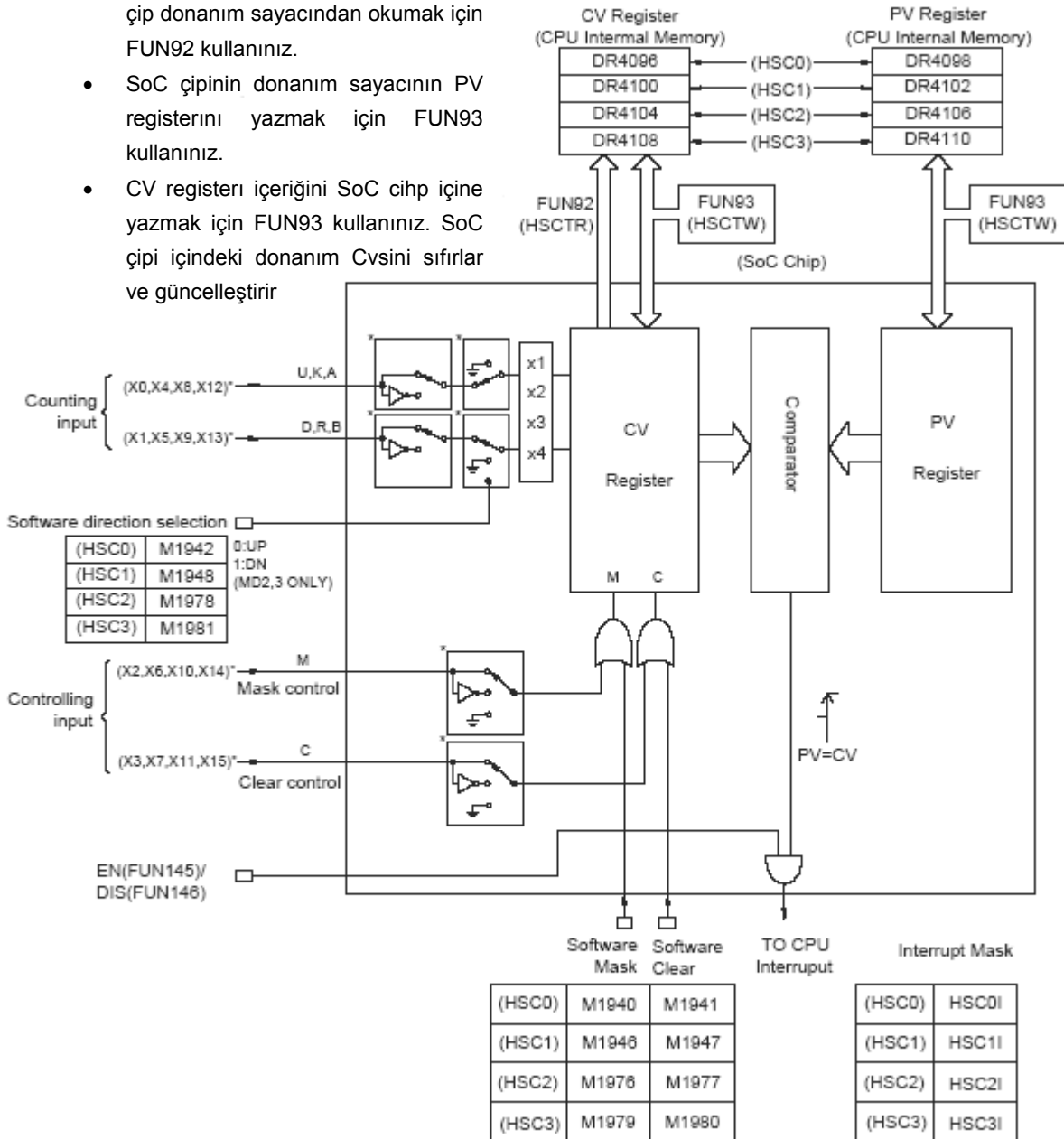
- Dalga şeklindeki pozitif /negatif kenar üzerindeki, yukarı/aşağı oklar (↑, ↓)sayımın nerede olduğunu gösterir. (+1 veya -1)

## 10.2 FBs- PLC Yüksek-Hızlı Sayıcı Sistem Yapısı

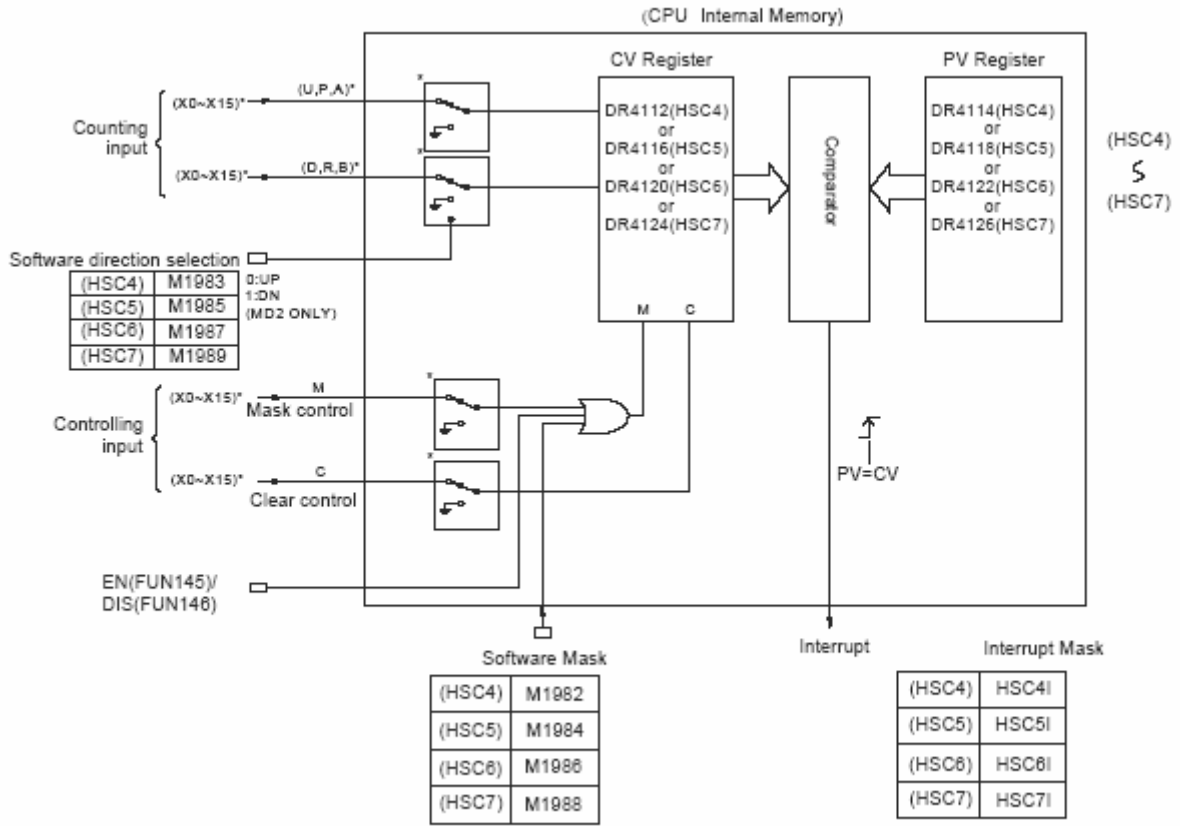
Aşağıda gösterilen çizelge FBs- PLC HHSC ve SHSC için sistem yapısıdır. Her biri çok amaçlı giriş ve sayım özelliklerine sahiptir. Bazı özellikler (CV register numarası, PV register numarası, interrupt etiketi ve MASK yazılımı için anahtar numarası, CLEAR ve yön seçimi gibi) kullanıcının yapılandırma için başvurma ihtiyacı duymayacağı şekilde yerleştirilmiştir. Ancak, aşağıdaki diyagramda “\*” ile işaretlenmiş bazı fonksiyonlarda HSC yapılandırması için programlama araçları kullanılmalıdır (HSC uygulaması seçimi, sayma modu, her fonksiyon giriş uygulaması, ters polarite ve giriş nokta numarası Xn'e uygun atama). Yapılandırmada ayrılan sayma modunun 8 çeşidinin işlevleri ve detaylı yapısı için bölüm 10.2.1~10.2.3'e bakınız.

Not: CV (güncel değer; PV (ayar değeri)

- Anlık sayma değerini CPU iç CV registeri içine koymak amacıyla, SoC çip donanım sayacından okumak için FUN92 kullanınız.
- SoC çipinin donanım sayacının PV registerını yazmak için FUN93 kullanınız.
- CV registeri içeriğini SoC cihp içine yazmak için FUN93 kullanınız. SoC çipi içindeki donanım Cvsini sıfırlar ve güncelleştirir



System Architecture of HHSC (HSC0~HSC3)



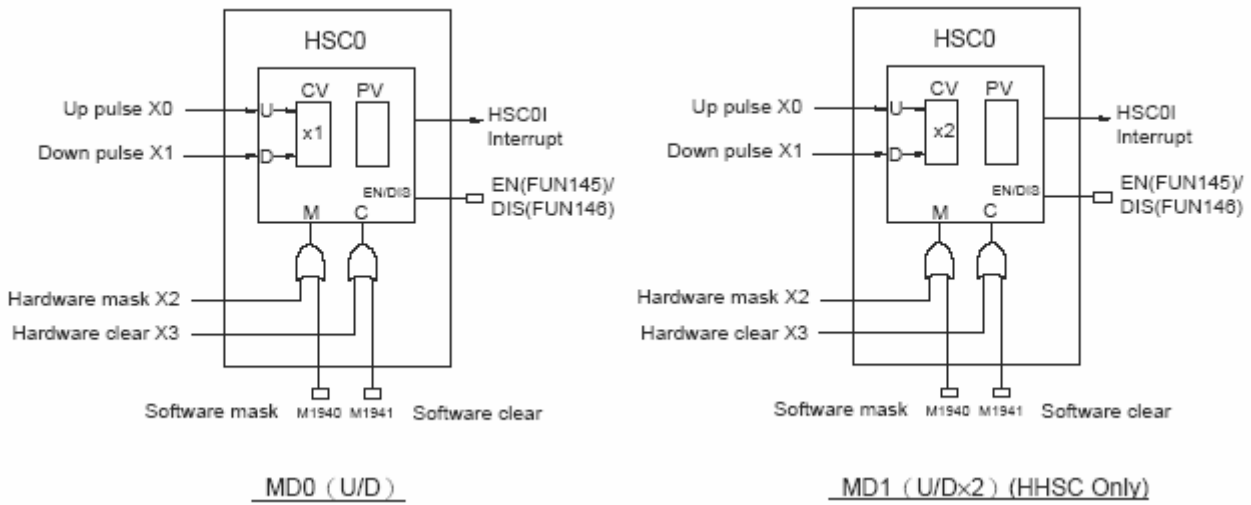
System Architecture of SHSC (HSC4~HSC7)

- HHSC ve SHSC nin tüm kontrol sinyalleri Active High olarak defaulttır (Örneğin; aktif için durum=1 ve pasif için durum=0). Sensör polaritesinin birlikte çalışması için HHSC sayma ve kontrol girişleri ters polaritede seçilmiş olabilirler.
- Varsayılan MASK kontrol sinyali, M=1 olduğunda, HSC sayma pulsi herhangi bir sayma gerçekleştirilmeden gizlenmiş olacak ve tüm HSC internal durumları değişmeden kalacaktır. M "0" olduğunda, HCS normal bir şekilde çalışmaya devam edecektir. Bazı sensörler, MASK'a göre enable çıkışlara sahiptir. Enable=0 olduğunda sayıcı saymayacak ve enable=1 olduğunda saymaya başlayabilir. Ayrıca, MASK'ın ters polarite giriş fonksiyonu, enable çıkışlı sensörler ile birlikte çalışacak şekilde seçilebilmelidir.
- CLEAR kontrol sinyali C=1 olduğunda, HSC internal CV registerı 0 olacak ve sayma işlemi gerçekleşmeyecektir. C=0'a döndüğünde, HSC 0' dan saymaya başlayacaktır. Ladder program, güncel sayma değerini 0 yapmak için direk olarak CV registerını (DR4116,DR4112,DR4120 ve DR4124) silebilir.
- FBs- PLC HHSC' nin dört seti SoC çiplerinde direk olarak yerleştirilmişlerdir. Burada, CV ve PV registerlarına kullanıcı direk olarak erişemez. Kullanıcı, CPU iç hafızasındaki CV registerına (DR4096~DR4110) erişebilir. İdeal durumda, çip içindeki CV ve PV registerlarının içerikleri, CPU iç hafızasındaki CV ve PV registerları ile aynı anda güncellenmelidir. Ancak, iki farklı devreye sahip olduklarında, ikisi arasındaki uygunluğu korumak için CPU tarafından kaydedilmiş veya yüklenmiş olmalıdırlar. CV ve PV registerlarını CPU içine ayrı ayrı yüklemek için FUN 93 kullanılması gerekir. (başlangıç değerinden saymaya başlamak için HHSC atanır). Sonra, CPU'daki CV registerına çiplerdeki HHSC CV registerının sayma değerini tekrar yüklemek için FUN 92 kullanılabilir (yani; CPU'daki CV registerı iki yönlü fonksiyona sahiptir). FUN 92 çalıştırıldığında yükleme uygulanabildiğinden, CPU'daki CV değeri ve çiplerdeki HHSC CV değeri arasında farklı sonuçlar olabilir. Bunlar; yüksek sayma frekansında özellikle çok büyük sapmalar şeklinde olacaktır.

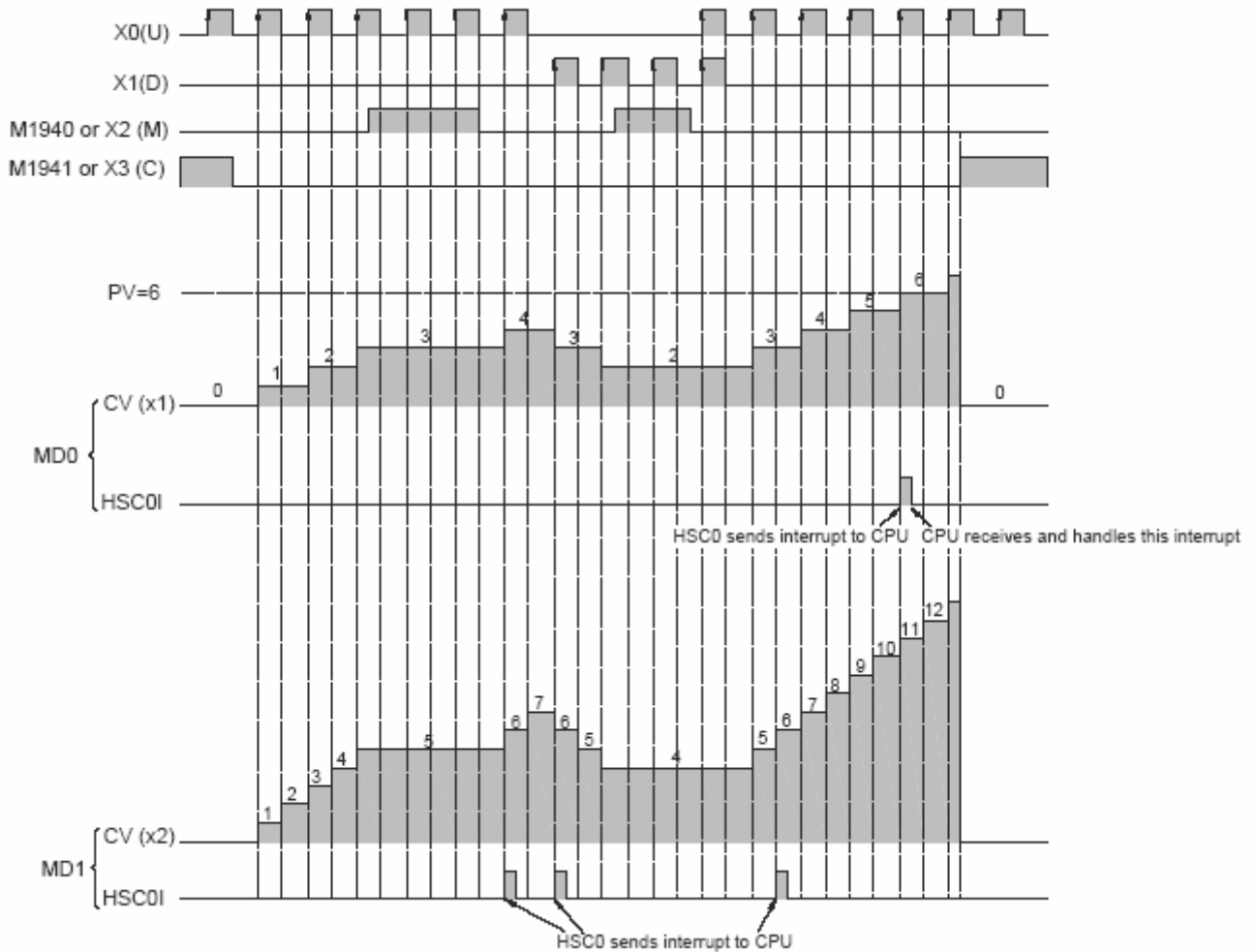
- Sayım frekansı düşük veya pozisyonlama hassaslığı az olduğunda, güncel sayma değerini kaydetmek için ana programdaki FUN 92 kullanılır ve o zaman birleşmiş karşılaştırıcı komutu basit sayma pozisyonlama kontrolü için elverişli olur.
- Pozisyonlama duyarlılığı ihtiyacı çok yüksek olduğunda veya çok bölgesel sayım ayar kontrolünde, daha kesin sayma pozisyonlama kontrolü gerçekleştirmek için birleşmiş karşılaştırma komutu ve interrupt programı zaman tabanında iken güncel sayma değeri kaydetmek için FUN 92 kullanılabilir.
- Pozisyonlama hassaslık gereksinimi çok yüksek olduğu sürece donanım sayıcısının belirlenmiş interrupt fonksiyonu kullanılmadığı. Ayar değeri çipteki HHSC'nin PV registerına FUN 93 tarafından yüklenebilir. HHSC CV değeri ayar değerine ulaştığında, HHSC'deki donanım karşılaştırıcı CV=PV olduğu anda CPU'ya interrupt gönderecek ve gerçek zamanlı kontrol veya işlem yapmak için interrupt alt programa geçecektir.
- Bir yandan SHSC, sayım girişi yükselen kenarda olduğunda CPU ya bir interrupt sinyali almak istediğinde, interrupt metodunu kullanır. Sonra CPU onun iç CV registerını azaltacağına veya arttıracığına karar verir(CPU içindeki CV registerının kendisi bir SHSC CV registerı olduğu için, FUN92 veya FUN93 gerekmemektedir).CV her güncellendiğinde, CPU onun PV register değerine eşit olduğunu bulursa CPU hemen uygulama için uygun SHSC interrupt programına atlayacaktır. SHSC sayımında veya kontrol girişinde bir değişiklik olduğu zaman bu CPU nun kesilmesine sebep olabilir. Sayım frekansı ne kadar büyükse o kadar çok CPU zamanını meşgul edecektir. CPU yanıtlama zamanı oldukça yükselecektir veya Watchdog zaman aşımı PLC yi işlem yürütmeyi durdurmaya zorlayacaktır. Bu sebepten, ilk önce HHSC nin kullanılması tercih edilir, eğer SHSC yi kullanma ihtiyacı duyulursa tüm FBs – PLC SHSC giriş frekanslarının toplamı 8 KHz e erişemez.
- MASK yazılımı, CLEAR ve yön kontrolü gibi özel röle kontrollerinin hiç biri gerçek zamanlı değildir. Dolayısıyla MASK, CLEAR veya yön değişimi rutin tarama esnasında ayarlanmış olmasına rağmen, I/O güncelleme rutin tarama tamamlandıktan sonra yapıldığında sinyal sadece HSC'ye iletilmiş olacaktır. Bundan dolayı, HSC işlemi gerçek zamanlı kontrol için uygun değildir (temel olarak HSC işlemi öncesi başlangıç ayarı için kullanılmalıdır). Gerçek zamanlı kontrole ihtiyaç duyulduğunda, giriş kontrolü için donanım kullanılması veya kontrol için FUN145 (EN),FUN146 (DIS), FUN92 (HSCTR) ve FUN93 (HSCTW) komutlarının uygulanması istenir.
- Her HSC ENable (FUN145) ve DISable (FUN146) fonksiyonlarına sahiptir. SHSC disable olduğunda interrupt fonksiyonu disable olmayacak ve sayma duracaktır. HHSC disable olduğunda ise saymaya devam edecek ama interrupt fonksiyonu disable olacaktır.

#### 10.2.1 Yüksek-Hızlı Sayıcının Yukarı/Aşağı Darbe Girişi Modu (MD0,MD1)

Yüksek hızlı sayıcının yukarı/aşağı pulse girişi, herhangi bir faz, birbirlerinden bağımsız yukarı darbe girişi (U) ve aşağı darbe girişine (D) sahiptir. Darbe girişi yükselen kenar olarak oluştuğunda, her bir CV değerinde +1 veya - 1 olacaktır (MD1 hem pozitif hem negatif kenar). Bu aynı zamanda eş zamanlı oluşan U ve D pulselerinin yükselen (veya düşen) kenarlarında uygulanır (birbirlerini dengeleyeceklerdir). Kontrol fonksiyonu kullanımda değilken durumu (M1940 ve M1941 gibi) "0" da kalmaması için iki mod için MASK ve CLEAR (CLEAR SHSC için uygun değildir) yerleşik yazılıma sahiptir. MASK kontrolü donanım ve yazılım kontrolünün OR işlemini öncelikli gerçekleştirir, sonra sonucu HSC MASK kontrol M 'ye gönderir. CLEAR da aynı şekilde çalışır. HSC'ı örnek alınacak olursa, MD0 ve MD1'in aynı şekilde yapılandırılması için şematik diyagram aşağıdaki gibidir.

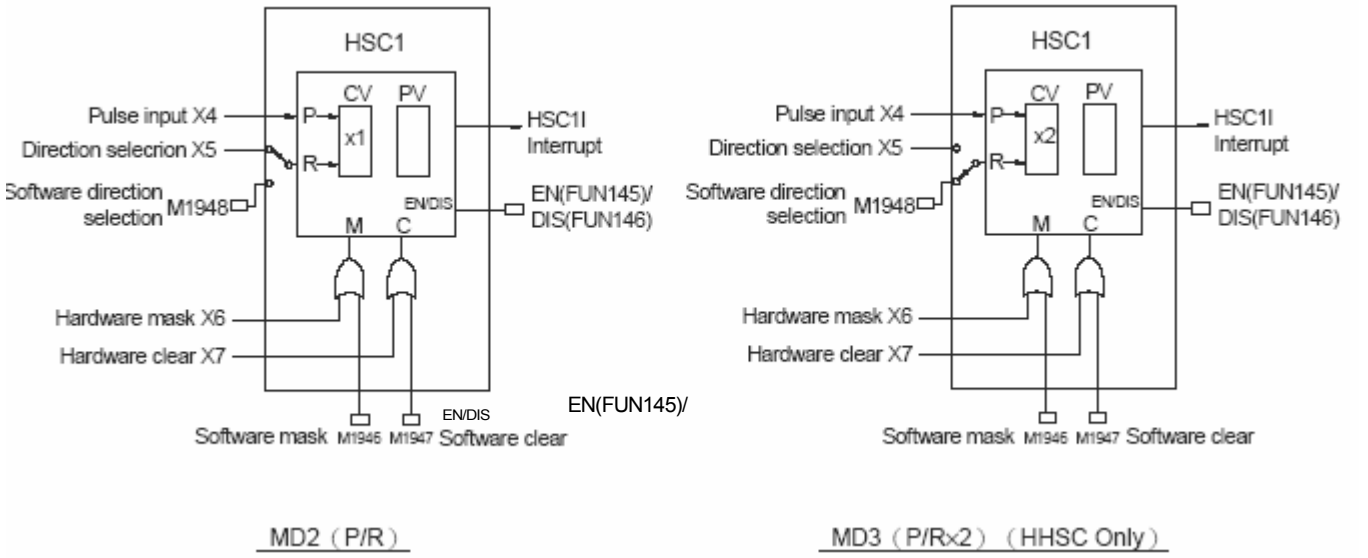


HSC nin Dalga şekilleri, yukarı / aşağı darbe giriş moduyla yapılandırılmış ve PV değeri 6'ya ayarlanmıştır:



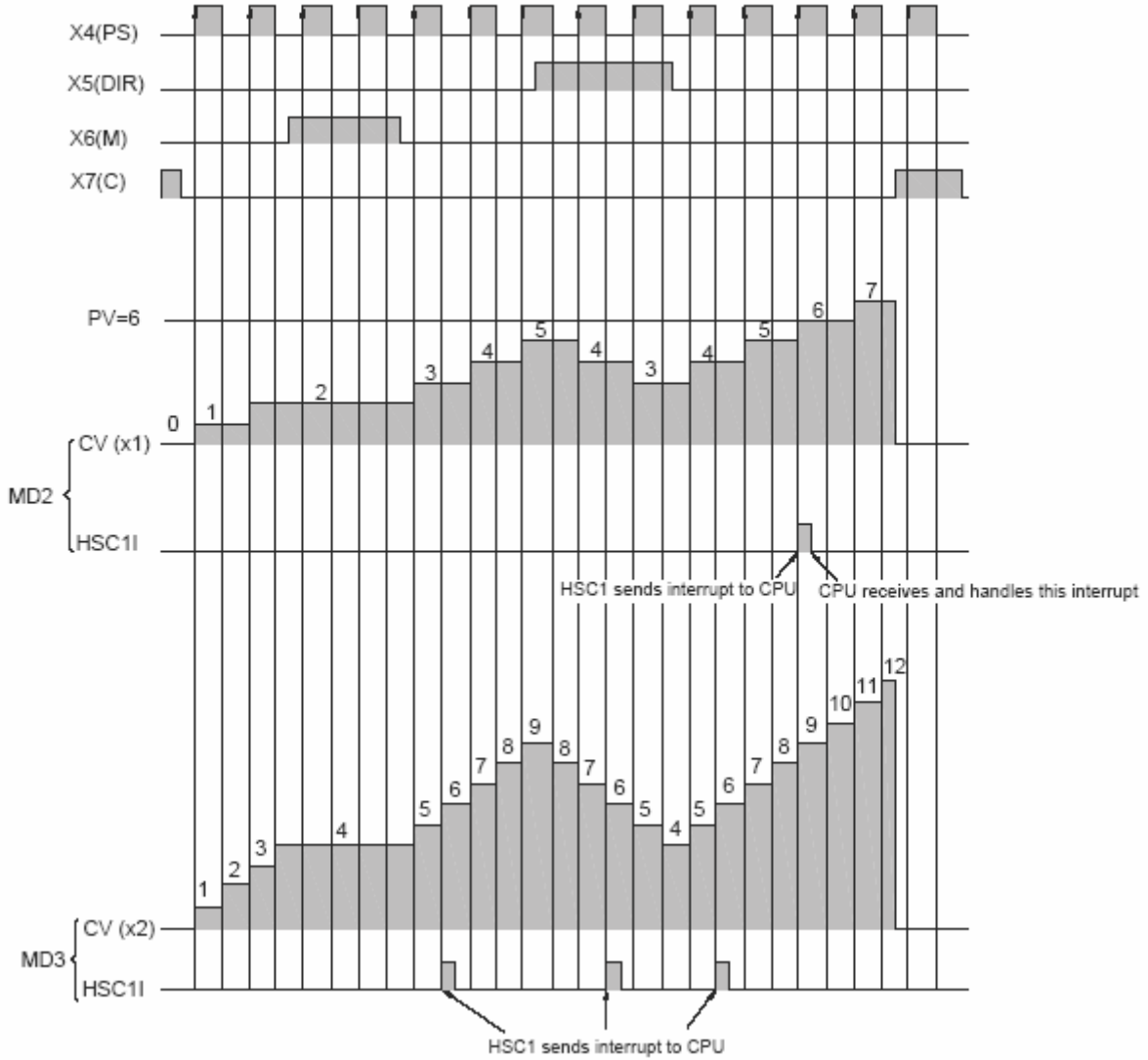
## 10.2.2 Yüksek-Hızlı Sayıcının Darbe/Yön Giriş Modu (MD2,MD3)

Darbe-yönlü giriş modlu yüksek-hızlı sayıcı sadece bir sayıcı pulse girişine P(pulse) sahiptir. Sayma pulsi yükselen kenara (MD3 için hem yükselen hem düşen kenarı ulaştığında CV değerinin +1 (R=0) ve -1 (R=1) olup olmadığına karar vermek için diğer yönde bir R girişi gerekir. Aynı durum MD2 ve MD3'ün sayması içinde uygulanır. MD2 PV değerini sadece yükselen kenarda sayarken, MD3 hem yükselen kenarda hem de düşen kenarda saymaktadır (MD2 iki kez sayar). Bu iki mod yerleşik olarak MASK ve CLEAR yazılımına sahiptir (SHSC' de CLEAR yoktur). Kontrol özelliği kullanımda olmadığına, 0 olması için durum korunmalıdır (örnekteki M1946 ve M1947 gibi). Yerleşik MASK ve CLEAR yazılımından ayrı olarak, MASK ve CLEAR donanım kontrolleri de yapılandırılmış olabilir. MASK kontrolü, donanım ve yazılım kontrolünün OR işlemini tarafından öncelikle gerçekleştirilmiştir. Sonuç HSC MASK kontrol M'ye gönderilir. CLEAR fonksiyonunda aynı şekilde çalışır. HSC1 fonksiyon şeması aşağıdaki şekilde gibi MD2 ve MD3 için yapılandırılmıştır;



HSC veya SHSC için, MD2 ve MD3 ün yön seçimi (harici giriş noktalarının kullanımını azaltmak için) harici girişlerden (bu örnekte olduğu gibi x5) veya CPU içindeki özel röleden (bu örnekte olduğu gibi M1948) gelebilir.

Aşağıdaki diyagram iki HSC nin sayım ve kontrol arasındaki ilişki için dalga şekli diyagramıdır.

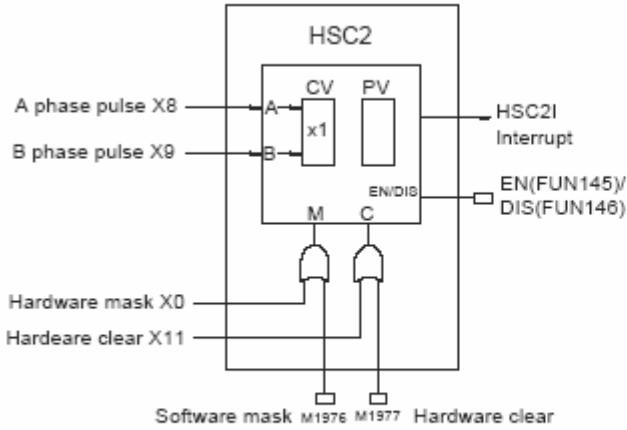


### 10.2.3 Yüksek-Hızlı Sayıcının AB Fazı Giriş Modu (MD4, MD5, MD6, MD7)

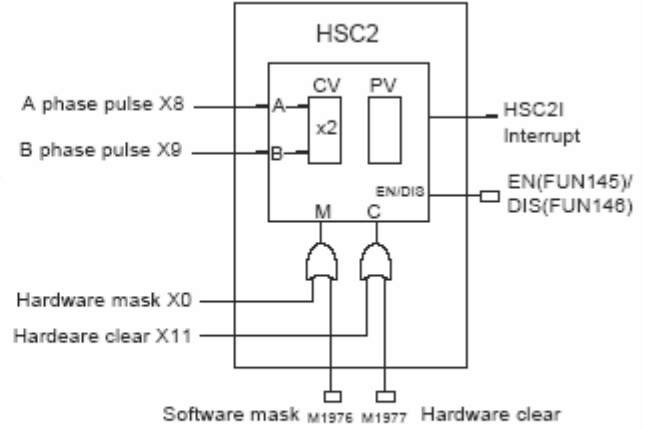
AB fazlı yüksek hızlı sayıcı, ikisi arasındaki ilişkiye bağlı olarak +1 veya -1 sayma değerli A ve B fazlı darbe girişleri ile donatılmıştır. Eğer A fazı, B fazının önünde ise CV değeri +1 olmalı, değilse -1 olmalıdır. AB fazı HSC nin dört modülünün, MD4(A/B),MD5(A/B x 2),MD6 (A/B x 3) ve MD/ (A/B x 4) sayımları benzerdir. Farklılıklar ise;

1. MD4 (A/B) : A, B' nin ilerisinde olduğunda A nın yükselen kenarı +1 ve A, B' nin gerisinde olduğunda A' nın düşen kenarı -1' dir.
2. MD5 (A/Bx2) : A, B' nin ilerisinde olduğunda, A' nın yükselen ve düşen kenarları +1 ve A, B' nin gerisinde olduğunda (MD4 iki kez sayılır) - 1' dir.
3. MD6 (A/Bx3) : A, B'nin ilerisinde olduğunda A'nın yükselen ve düşen kenarları ve B' nin yükselen kenarı +1 dir. A, B' nin gerisinde olduğunda (MD4 üç kez sayılır) A' nın yükselen kenarı ve B' nin düşen kenarı -1 dir.
4. MD7 (A/Bx4) : A, B' nin ilerisinde olduğunda A ve B nin yükselen ve düşen kenarları +1 dir ve A, B' nin ilerisinde olduğunda A' nın ve B' nin düşen kenarı -1 dir.

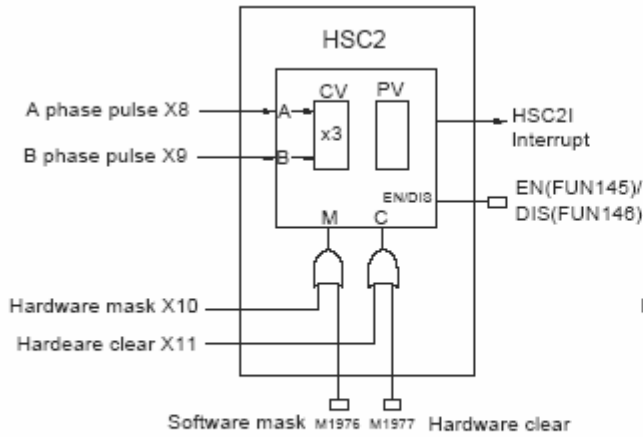
Diğer MD4 ~ MD7 HSC modları yerleşik olarak MASK ve CLEAR (SHSC, CLAER'a sahip değildir) yazılımlarına sahiptirler. Kontrol fonksiyonu kullanılmadığında, 0 olması için durum korunmalıdır (Bu örnekte M1946 ve M1947 gibi). Yerleşik MASK VE CLEAR yazılımlarından ayrı olarak, MASK ve CLEAR donanımlarının kontrolleri yapılandırılabilir. MASK kontrolü, donanım ve yazılım kontrolünün OR işlemini öncelikle gerçekleştirecek, sonra sonuç HSC MASK kontrol M'ye gönderilir, CLEAR fonksiyonunda aynı şekilde çalışır. Dört MD4 ~ MD7 HSC modu için HSC2 nin şematik özellikteki diyagramları aşağıdaki gibidir;



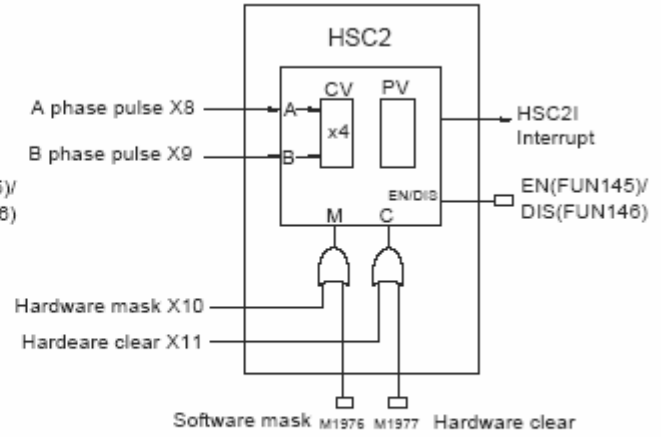
MD4 (A/B)



MD5 (A/Bx2) (HHSC Only)

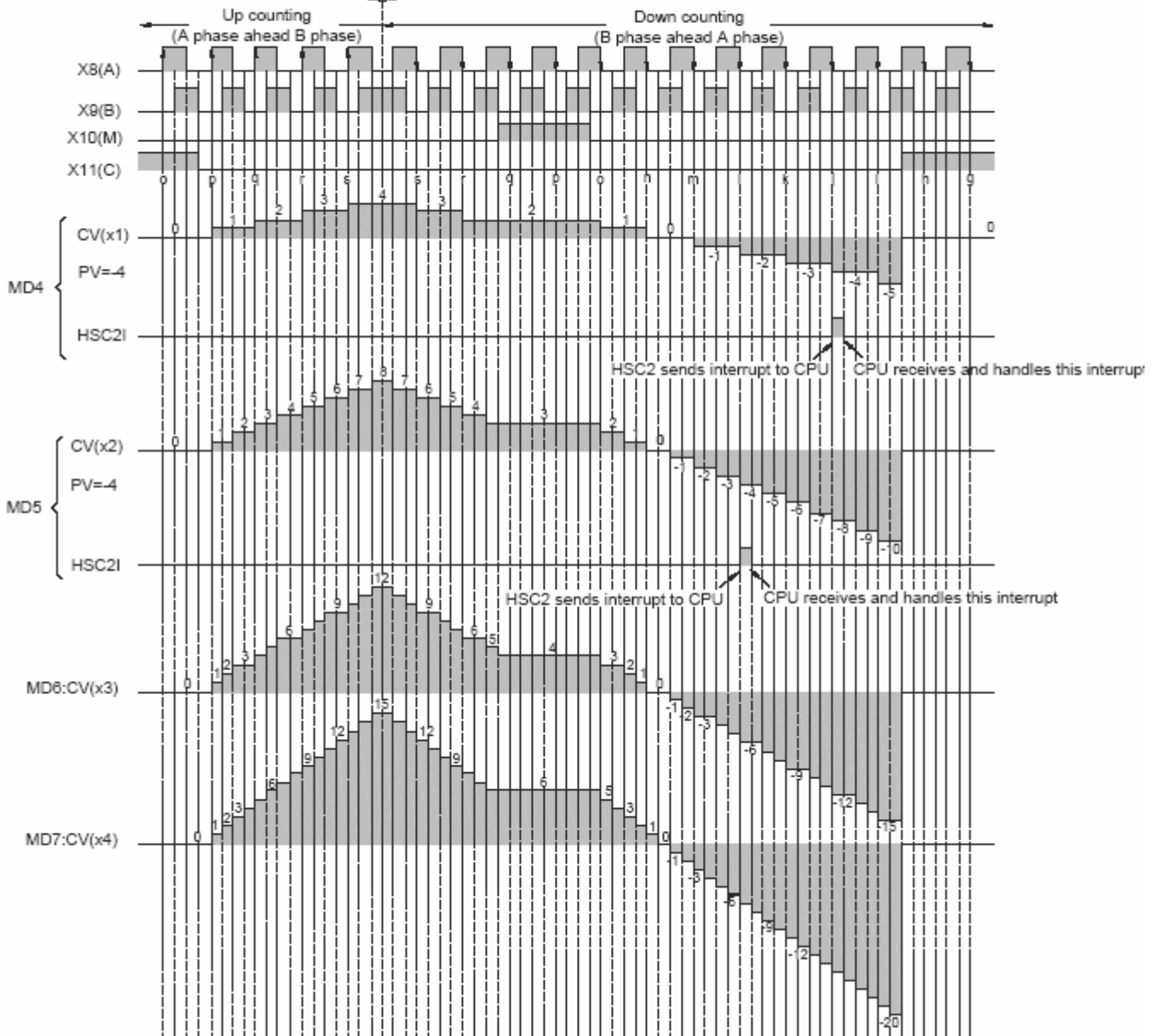
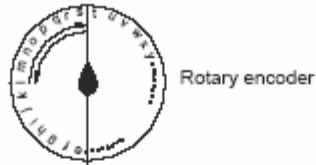
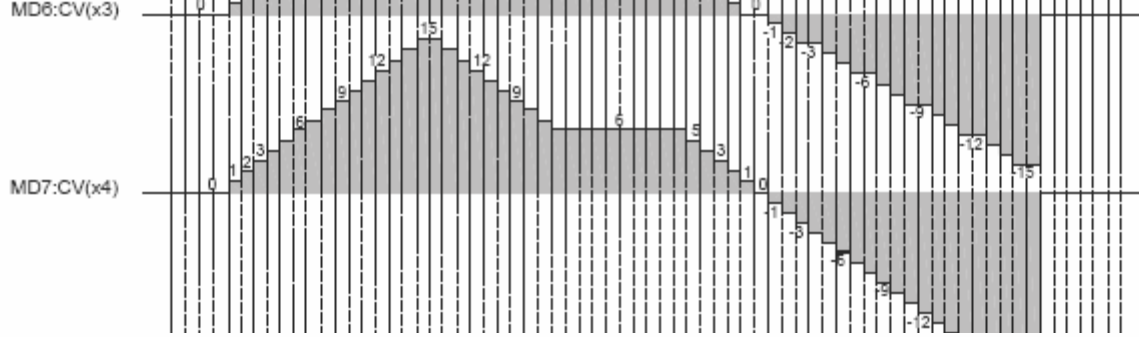


MD6 (A/Bx3) (HHSC Only)

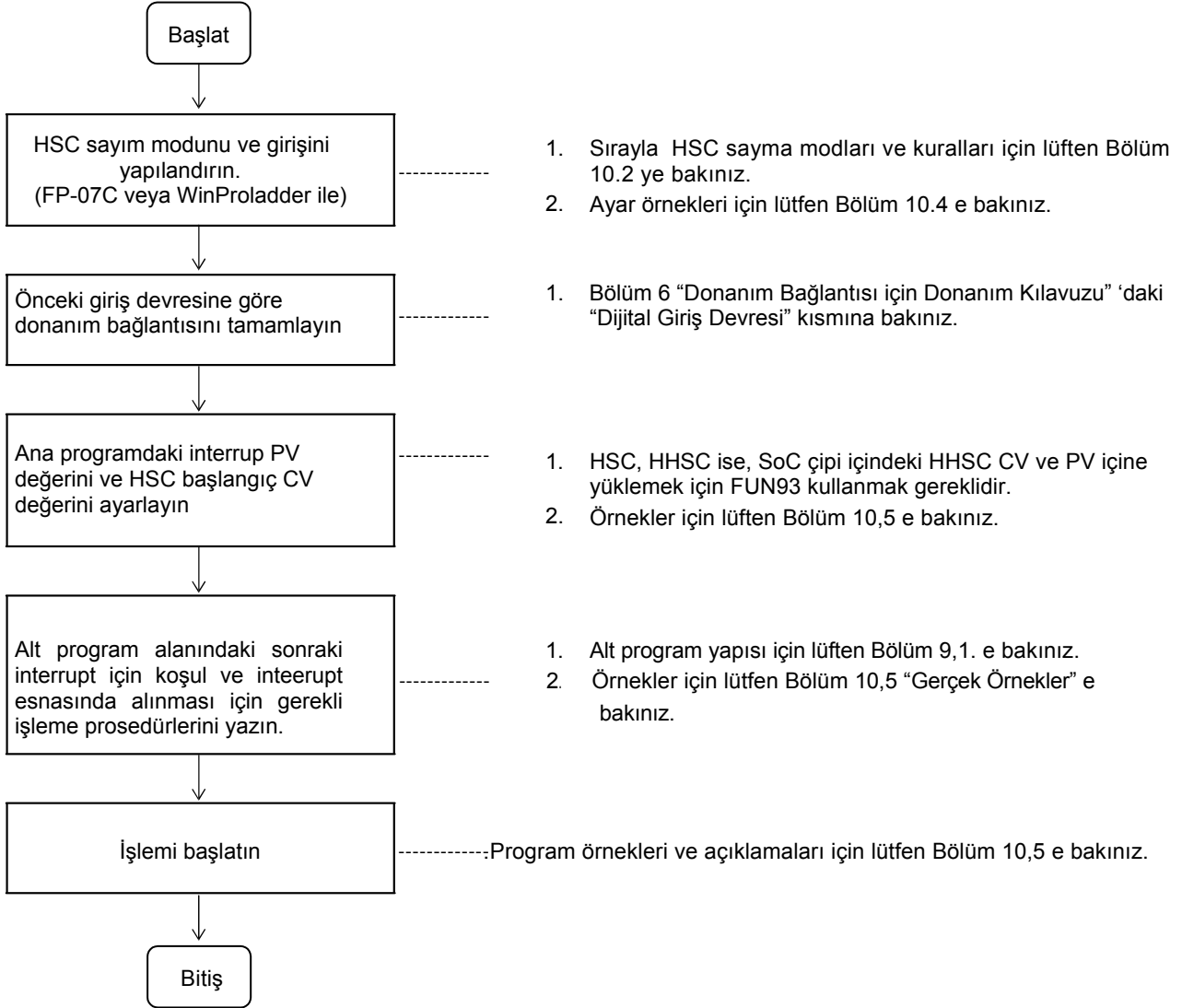


MD7 (A/Bx4) (HHSC Only)





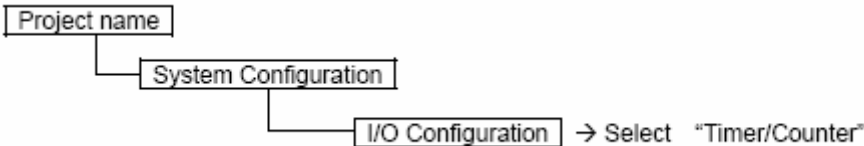
## 10.3 FBs- PLC Yüksek-Hızlı Sayıcı Uygulama Prosedürü



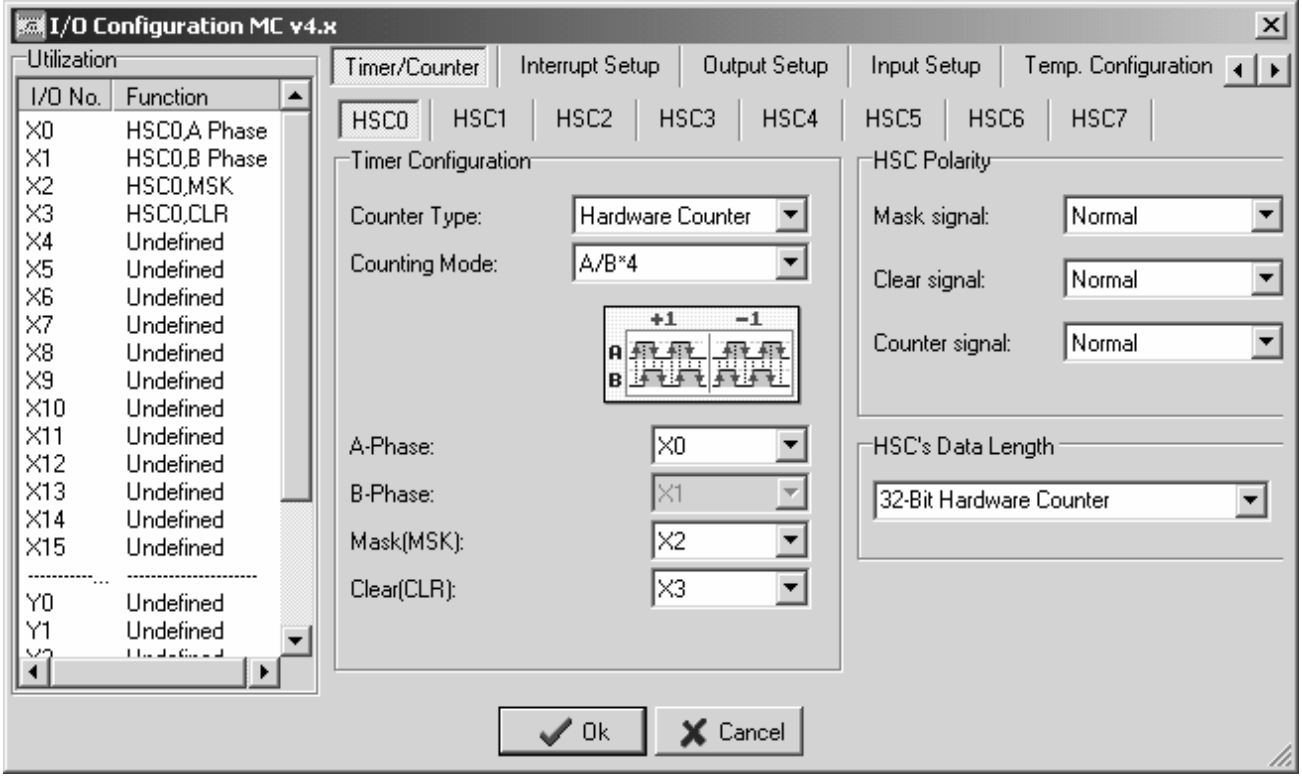
## 10.4 HSC/HST Yapılandırması

### 10.4.1 HSC/HST Yapılandırması (WinProladder Kullanarak)

Project penceresindeki "I/O Configuration" seçeneğine tıklayınız:



"Zamanlayıcı/Sayıcı" penceresi görüldüğünde istenilen Zamanlayıcıyı veya Sayıcıyı seçebilirsiniz.



--- 《Zamanlayıcı/Sayaç Yapılandırması》 ---

【 Sayıcı Tipi 】 : Donanım Sayıcısı veya Donanım Zamanlayıcısı seçilebilir.

【 Sayma Modu 】 : Sayma Modu ( Örnek: U/D、 P/R、 A/B...)seçilebilir

【 A-Fazı 】 : Yukarı pulse giriş sinyalini seçin. Eğer Mod, P/R sayma Modunda ise bu öge “PS” olacaktır; eğer Mod U/D sayma Modunda ise bu öge “Up” olacaktır.

【 B-Fazı 】 : Aşağı pulse giriş sinyalini seçin. Eğer Mod, P/R sayma modunda ise bu öge “DIR” olacaktır; eğer mod, U/D sayma modunda ise bu öge “Dn” olacaktır.

【 Mask[MSK] 】 : Mask girişi seçilebilir.

【 Clear[CLR] 】 : Clear girişi seçilebilir.

---HSC Polarite Alanı---

【 Mask sinyali 】 : Mask sinyalinin pozitif veya negatif olmasını belirler.

【 Clear sinyali 】 : Clear sinyalinin pozitif veya negatif olmasını belirler.

【 Counter sinyali 】 : Counter sinyalinin pozitif veya negatif olmasını belirler.

--- 《HSC'nin Data Uzunluk Alanı》 ---

32 bit donanım sayıcı modu veya 16 bit zamanlayıcı + 16 bit sayıcı modu seçilebilir. 32 bit Donanım Sayıcı modu sayma değerini kaydetmek için iki register kullanılması anlamına gelir. 16 bit Zamanlayıcı + 16 bit Sayıcı Modu sayma değerini kaydetmek için bir register ve bir dairesel zaman registerı kullanılmasıyla oluşturulur.

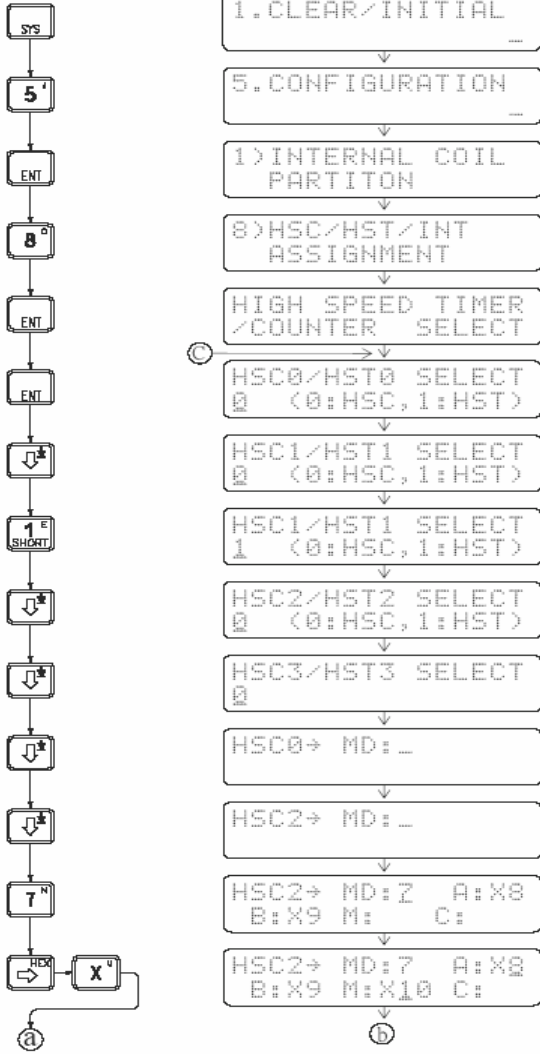
#### 10.4.2 HSC/HST Yapılandırması (FP- 07C kullanarak)

FP-07C ekranı, bu bölümde HSC yapılandırmasını açıklamak için örnek olarak kullanılacaktır. HSC Yapılandırması, sırasıyla aşağıdaki 5 ögeyi içermektedir;

1. HSC/HST için görev seçin (Sadece HHSC bu öge seçimi özelliğine sahiptir). Seçim HSC ise bir sonraki ögeye ilerler. HST olarak yapılandırıldıysa başka ögeye gerek yoktur.
2. HSC sayma modlarını ayrı ayrı görevlendirin (MD0 ~ MD7). Mod numarası girildikten sonra, FP-07 otomatik olarak HSC sayımını ve modun kontrol giriş isimlerini ve harici giriş nokta sayısı Xn tuşlamak için kullanıcıya ayrılmış alanı gösterecektir. Boşluk mod alanı HSC nin kullanımda olmadığını gösterir.
3. Sırasıyla sayma girişleri (U, D, P,R, A ve B) ve kontrol girişlerinin (M ve C) uygulanıp uygulanmayacağını belirleyin (kullanımda değilse alanı ayırın ve uygulanmış olan Xn değeri ile doldurun. Çünkü HHSC'nin ayrı Xn giriş değerleri sabittir. Sadece "X" harfini tuşlamak gerekir ve FP-07C otomatik olarak ayarlanmış n sayısını yazacaktır).
4. Kodlayıcının polaritelerini karşılaştırmak amacıyla, HHSC sayma girişlerinin (U, D, P, R, A ve B) ters olup olmayacağını seçin (0: ters değil, 1: ters. Önceden 0 ayarlanmıştır).
5. Kodlayıcının polaritelerini karşılaştırmak amacıyla, HHSC kontrol girişlerinin (M ve C) ters olup olmayacağını seçin (0: ters değil, 1: ters. Önceden 0 ayarlanmıştır).

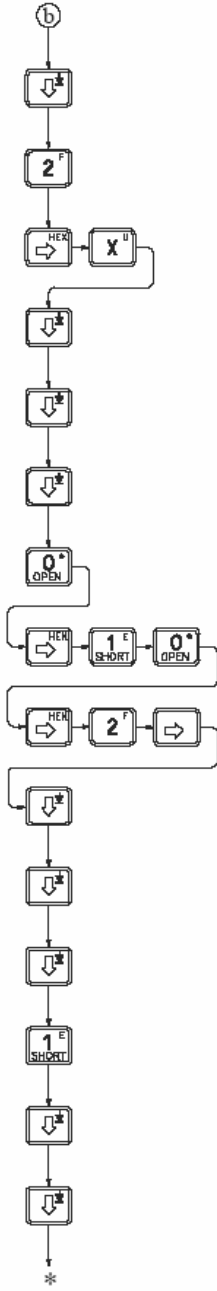
【Anahtar Uygulaması】

【LCD Görüntüsü】

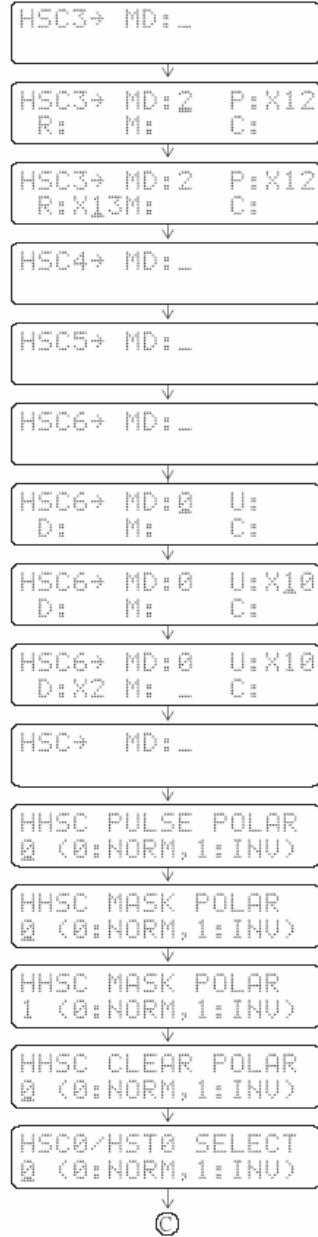


- ※ HSC0,HSC olarak belirlenmiştir
- ※ HSC1,HSC olarak belirlenmiştir
- ※ HSC1 yüksek hızlı sayıcı HST1 olarak kullanılmıştır
- ※ HSC2, HSC olarak belirlenmiştir.
- ※ HSC3, HSC olarak belirlenmiştir.
- ※ MD alanı HSC0 kullanımında olmadığına boş gösterilir.
- ※ HSC1, HST olarak ayarlandığından yapılandırmaya gerek yoktur (Görüntülenmez).
- ※ HSC2' nin MD alanı MD nin kullanımında olmadığına boş gösterilir.
- ※ \*7 tuşlandıktan sonra, MD7 isimleri görüntülenecek ve A ve B için ayar değerleri otomatik olarak doldurulacaktır. (X8 veya X9)
- ※ MASK gerekliyse, sadece X tuşlanacak ve otomatik olarak 10 numarası yazılacaktır.

## Anahtar Uygulaması]



## LCD Görüntülemesi



- \* 2 'yi tuşlayın ve P'nin önceden belirlenen numarası otomatik olarak gelecektir.
- \* İmleci "R" ye hareket ettirin ve "X"i tuşlayın, hemen X13'e dönüşecektir ve onu donanım yönü kontrolü olarak şekilde değiştirecektir.
- \* HSC4 kullanımda değildir.
- \* HSC5 kullanımda değildir.
- \* HSC6,MD0 olarak ayarlanmıştır ve MD0 SHSC nin ayrı ayrı giriş isimleri otomatik olarak görüntülenecektir.
- \* HSC6 için, X10 u yukarı sayım darbeu "U" olarak görevlendirilir..
- \* HSC6 için X2 yi aşağı sayım darbeu "D" olarak atayın.
- \* HSC7 kullanımda değildir.
- \* Her HHSC'nin (HSC0 ~ HSC3) tüm sayma girişlerini önceden terslemez
- \* HHSC tüm MASK kontrolleri önceden terslemez
- \* HHSC nin tüm MASK girişlerini ters olarak değiştirir. (örneğin:MASK özelliği enable olur)
- \* HHSC nin tüm CLEAR kontrolleri Önceden terslemez.
- \* Yapılandırmayı tamamlar ve başlangıç ekranı görüntüsüne geri döner. (HSC0/HST0 öge seçimi)

- \* Giriş değeri değişikliği direk olarak yeni değer girilerek yapılabilir. Gerekirse, herhangi bir giriş değerini silmek için "CLR" kullanın.
- \* Boş alan (hiçbir değer yüklenmemiş) HSC nin uygulamasını veya girişin gerekmediğini gösterir..
- \* Bir önceki örnekteki "darbe" "Sayma Girişi" ni gösterir. Örneğin HHSC nin U, D, P, R veya A ve B si.
- \* "POLAR" "POLARITY" yi temsil eder. Örneğin, ters veya ters-değil seçimi

⌘ HHSC sayma ve kontrol girişleri için giriş noktaları sabittir. Bu yüzden, önceki örneğin "Yapılandırma Örnekleri"nde, FP-07C veya Winproladder X için önceden belirlenen sayıyı otomatik olarak oluşturacak ve girişe uygulamış olacaktır. Bunu göstermek için herhangi bir HHSC girişinde "X" tuşlamak gerekir. Kullanıcı, SHSC sayma veya kontrol girişlerini X0~X15 aralığında rastgele atayabilir. Ancak, işlemin tamamlanması için SHSC giriş nokta numarasına uygun "X" ve n sayısının girilmesi gereklidir.

Tüm önceden belirlenmiş veya enable giriş noktası numaraları ,MASK yazılımı, CLEAR yazılımı, yön seçimi ve HHSC nin ve SHSC nin diğer ilişkili numaraları aşağıdaki taboda gösterilmiştir.

Signal Allowed		Type	MC/MN							MA	
			HHSC				SHSC			SHSC	
			HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	HSC6	HSC7	HSC4 ~ HSC7
CV Register			DR4096	DR4100	DR4104	DR4108	DR4112	DR4116	DR4120	DR4124	The same as SHSC of MC/MN
PV Register			DR4098	DR4102	DR4106	DR4110	DR4114	DR4118	DR4122	DR4126	The same as SHSC of MC/MN
Counting Input	U,P or A		X0	X1/X4	X4/X5/X8	X5/X12	X0~X15	X0~X15	X0~X15	X0~X15	The same as SHSC of MC/MN
	D,R or B		X1	X5	X9	X13	X0~X15*	X0~X15*	X0~X15*	X0~X15*	The same as SHSC of MC/MN
Control Input	Mask		X2	X6	X10	X14	X0~X15	X0~X15	X0~X15	X0~X15	The same as SHSC of MC/MN
	Clear		X3	X7	X11	X15	X0~X15	X0~X15	X0~X15	X0~X15	The same as SHSC of MC/MN
Software MASK Relay			M1940	M1946	M1976	M1979	M1982	M1984	M1986	M1988	The same as SHSC of MC/MN
Software CLEAR Relay			M1941	M1947	M1977	M1980	Clear the Current Value Register directly				
Software Direction Selection(MD2,3 Only)			M1942	M1948	M1978	M1981	M1983	M1985	M1987	M1989	The same as SHSC of MC/MN
Interrupt Subroutine Label			HSC0I	HSC1I	HSC2I	HSC3I	HSC4I	HSC5I	HSC6I	HSC7I	The same as SHSC of MC/MN

\* SHSC MD2' de (P/R) çalıştırıldığında, yön seçimi özel röleler M1983,M1985,M1987 ve M1989 ile yapılır.

⌘ A-B Modunda (HHSC, MD4 ~ MD7 olarak; SHSC MD4 olarak) çalışırken A/B girişi X8 ve X9 gibi çift olarak kullanılmalıdır (çift numara A fazı ve tek numara B fazı).

⌘ Yukarıdaki tablodaki X0 ~ X15 giriş noktası sadece bir kez atanabilir ve tekrar kullanılamaz. (Örneğin; tek fonksiyon olarak kullanılabilir.)

⌘ FBs-MN'nin frekansı 920 KHz ye ulaşabilir. (tek faz ve AB fazı)

⌘ FBs-MC nin frekansı 120 KHz ye ulaşabilir. (tek faz ve AB fazı)

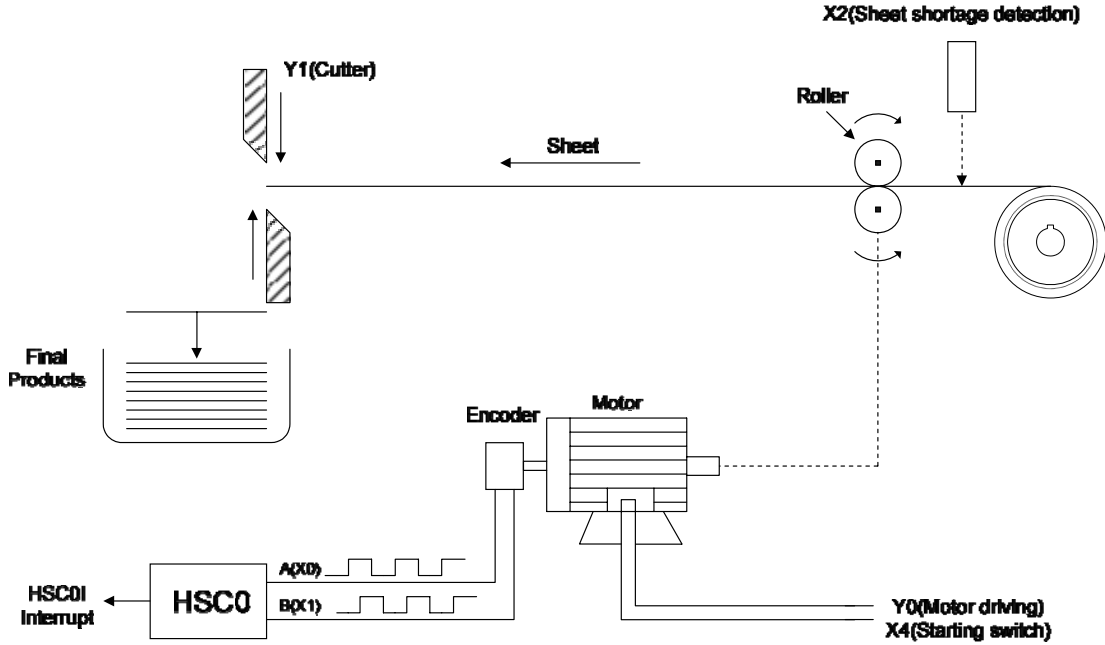
⌘ SHSC'nin toplam giriş frekansı 8 KHz'i aşamaz, frekans ne kadar yüksek ise sistem CPU süresini o kadar meşgul eder ve tarama süresi aniden genişler.

⌘ MA sadece SHSC'yi destekler.

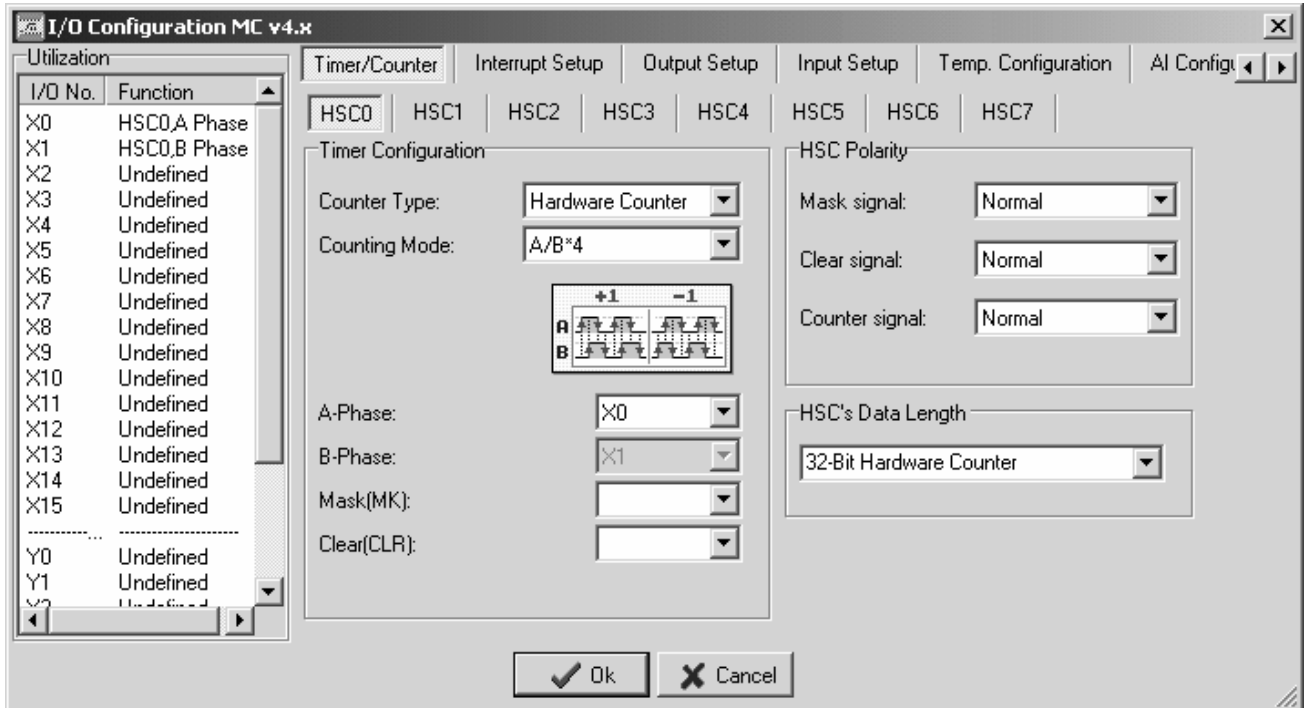
## 10.5 Yüksek Hızlı Sayıcı Uygulaması İçin Örnekler

**Örnek 1** Bu örnekte, eşit-genişlikteki kesim kontrolü için yüksek-hızlı sayıcı kullanılmıştır.

### Mekanizma



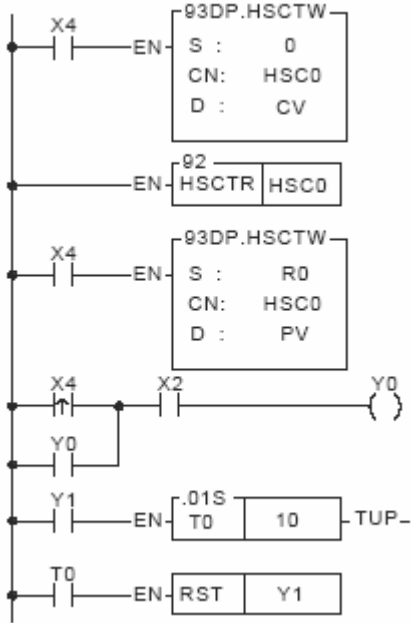
**HSC yapılandırması** (Sadece MD7' yi HSC0' a kurun ve yapılandırmayı tamamlayın)





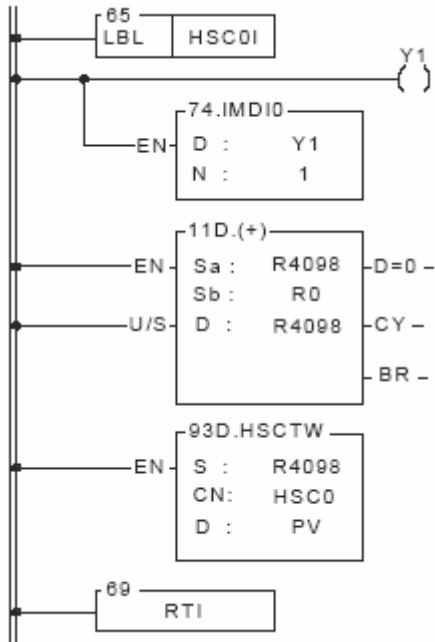
## Kontrol programı

### Ana program



- SoC çipinde HSC0'ın CV registerı içine güncel register değerinin içeriğini yazmak için FUN 93 kullanılır.  
CN= 0 olduğunda HSC0 gösterilir  
D= 0 olduğunda CV gösterir.
- SoC çipi içindeki HSC0' ın sayma değerini okumak için FUN 92 kullanılır. (DR4096 içiner depolar.)
- DR4098 içine DR0 kesim boyutunu sayarak depolar ve SOC çipinde HSC0'ın PV registerı içine değeri depolamak için FUN93 kullanılır.  
CN =0 olduğunda HSC0'ı gösterir.  
D =1 olduğunda Pv' yi gösterir.
- Motoru çalışır.
- 0,1 saniye için Y1 ON olur. Kesici çalışır.

### Alt Program



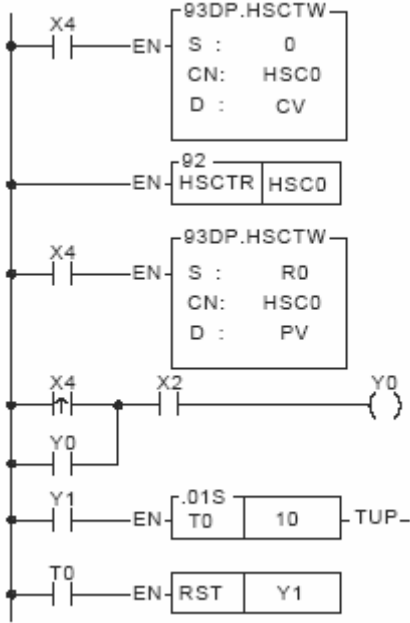
- SoC çipinde HSC0 CV=PV olduğunda, HSC0I şeklinde etiketlenmiş interrupt alt programı otomatik olarak çalışacaktır
- Sayım yukarı doğru olduğunda, Y1 ON olur. (malezeler kesilir)
- Tarama zamanının sebep olduğu hatayı azaltmak için çıkış Y1 olur.
- Yeni kesme pozisyonunu ve HSC0 PV hesaplar.

### 【Açıklama】

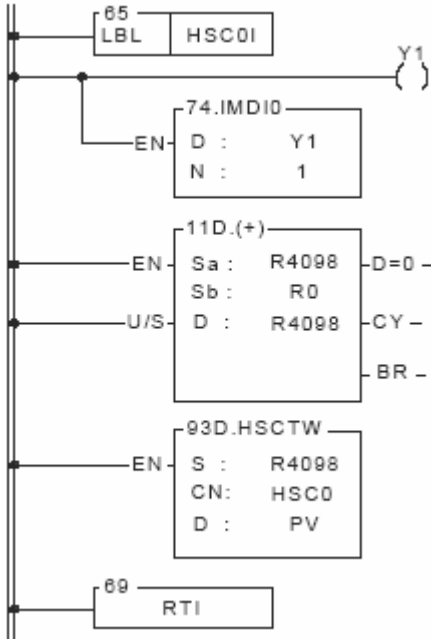
1. Malzeme taşınması motor Y0 konumuna gelince başlar. Ana program önceden HSC0 CV (CV=0) başlangıç durumuna geçecektir ve kırpma uzunluğunu (DR0) HHC0 PV'ye taşır.
2. CV, PV'ye ulaştığında, R0' ın uzunluğu HSC0 PV içine tekrar yüklenmeden önce PV eklenmiştir.
3. Tüm materyalleri taşındığında, malzeme eksik dedektörü X2, ON olacak ve motoru durduracaktır.

**Örnek 2**

Interruptla ile işlenmiş yüksek hızlı yukarı sayma hareketi örneği

**[Ana Program]**

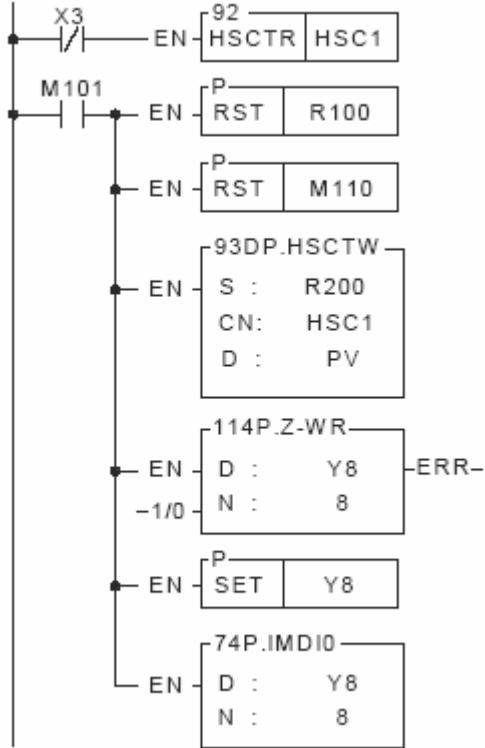
- Soc' de HSC0 CV içine güncel register değerinin içeriğini yazmak için FUN 93 kullanır .  
CN =0 olduğunda HSC0' ı temsil eder.  
D =0, CV yi temsil eder.
- SoC çipinde HSC0 PV içine önceden belirlenen register içeriğini yazmak için FUN93 kullanır ve bunu CV registerına depolar (DR4096).  
CN=0, HSC0' ı temsil eder.
- M101 0→1 olduğunda, Y0 ON olur ve çalışmaya başlar
- SoC çipinde HSC0 PV içine önceden belirlenen registerın içeriğini yazmak için FUN93 kullanır. Bu yukarı sayma interruptının ayar değeri sunar.  
CN=0, HSC0 gösterilir  
D =1, PV gösterilir.

**[Alt Program]**

- Donanım yüksek hızlı sayıcı #0 interrupt etiketi
- Time up olduğunda, Y0 OFF olur (durur).
- Hemen durdurmak için Y0 çıkışına izin verir.  
(aksi takdirde Y0 da tarama zamanı çıkış gecikmesi oluşturacaktır.)

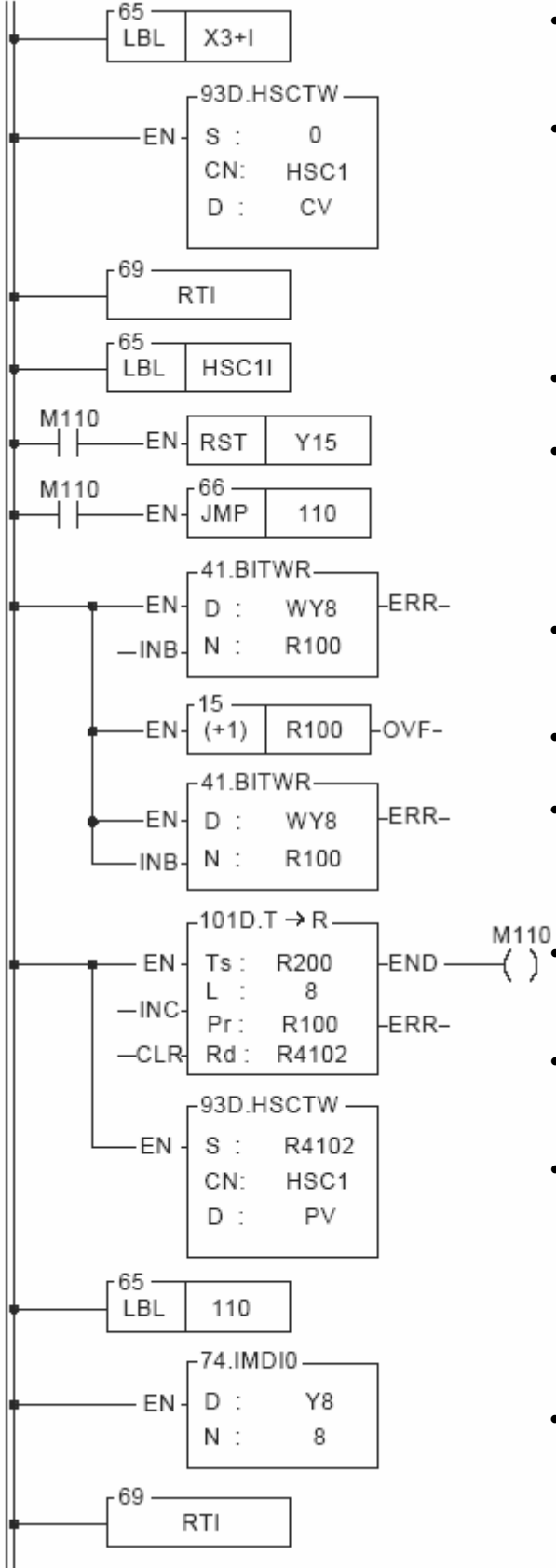
**Örnek 3**

Interrupt işleminde çok bölge yüksek hızlı yukarı sayma ani cevap örneği

**【Ana program】**

- SoC çipinde HSC1 güncel değeri okumak için FUN 92 kullanılır ve DR4100 güncel register değeri içine depolar. CN =1 olduğunda HSC1 gösterilir
- M101 0→1 olduğunda, gösterge registerı sıfırlanır
- Son bölge bayraklarını OFF yapar.
- SoC'de HSC1 PV içine önceden belirlenen register içeriğini yazmak için FUN 93 kullanılır CN =1 olduğunda HSC1 gösterilir D =1 olduğunda Pv gösterilir
- Y8~ Y15 aralığı OFF olur
- Y8 i ON yapar, bölgeyi 0'da gösterir.
- Y8~Y15 çıkışını ayarlar.

【Alt Program】



- X3+ 1' in X3 artan kenar interrupt servisi alt yazılımı için etiket adı  
(artan kenar interrupt girişi olması için X3 atanmalıdır)
- X3 0→1'e değiştiğinde, SoC çipi (sıfırlama) içindeki HSC1 CV' ye güncel register içeriğini yazmak için FUN93 kullanır. CN = 1olduğunda HSC1gösterir. D = 1olduğunda CV' yi gösterir
- HSC1I yüksek hızlı donanım sayacı interrupt servisi alt programı olarak etiketlenmiştir
- Son bölge tamamlandığında Y15 OFF olur.
- Önceki bölge çıkışını OFF yapar.
- Gösterge noktasını sonraki bölgeye ayarlar.
- Bir sonraki bölümün çıkışını ON yapar.
- Sonraki bölgenin sayma değerini önceden belirlenmiş DR4102 registerına taşır (DR200 göstergesinden başarak register işaretlenmiştir)
- Sonuncu bölgede; M110, ON olur.
- SoC çipi içinde HSC1 PV içine önceden belirlenmiş değeri yazmak için FUN93 kullanır, yukarı sayma interrupt ayar noktası gibi çalışır. CN =1olduğunda HSC1 gösterir. D =1olduğunda PV gösterir.
- Y8 ~ Y15 çıkış hemen iletilir.

## 10.6 FBs-PLC Yüksek-Hızlı Zamanlayıcı

Sıradan bir PLC nin minimum zamanlama ünitesi (zaman tabanı) sadece 1 ms' ye ulaşabilir. Tarama zamanındaki sapma da buna eklenmelidir. Bu yüzden, daha kesin zamanlama isteniyorsa yüksek hızlı zamanlayıcı (HST) kullanmak gerekir (örneğin frekans ölçümü için HSC ile birlikte zamanlayıcı kullanılır).

FBs- PLC 16 bit/0.1 ms zaman tabanlı bir yüksek hız zamanlayıcıyla (HSTA) ve daha önce açıklandığı gibi kullanım için 32 bit/0.1 ms zaman tabanı ile yüksek hız zamanlayıcı olarak çalışan, HHSC nin dört 32-bit yüksek hızlı sayacı (HST0~HST3) ile yapılandırılmıştır. Bu nedenle, FBs- PLC beş yüksek hızlı zamanlayıcıya sahip olabilir. HSC ve INT gibi tüm HST' ler EN (FUN145) ve DIS (FUN146) komutları tarafından enable veya disable edilebilirler. HSTA ve HST0 ~ HST3 aşağıda sırasıyla açıklanmışlardır.

Çoğu sıradan PLC için en iyi zaman tabanı 10ms dir. Yine de bazı PLC' ler 1 ms zaman tabanlı HST' ye sahip olabilirler. PLC tarama zamanındaki sapmalar dikkate alındığında (örneğin; eğer tarama süresi 10 ms içse, zaman tabanı 1 ms iken, toplam sapma hala 10 ms i aşmıştır) 1 ms yapılandırması anlamsızdır. Bu yüzden PLC ler yüksek kesinlikteki zamanlamada çalıştırılmazlar. 0,1 ms zaman tabanlı FBs\_PLC nin süre dolumu olağan PLC nin zamanlama uygulamasından 100 kez daha iyi bir kesinlikle saptamak amacıyla interrupt tarafından gönderildiği için, tarama süresi içinde sapması yoktur ve kesin zamanlamaya dayanarak birçok uygulama için kullanılabilir.

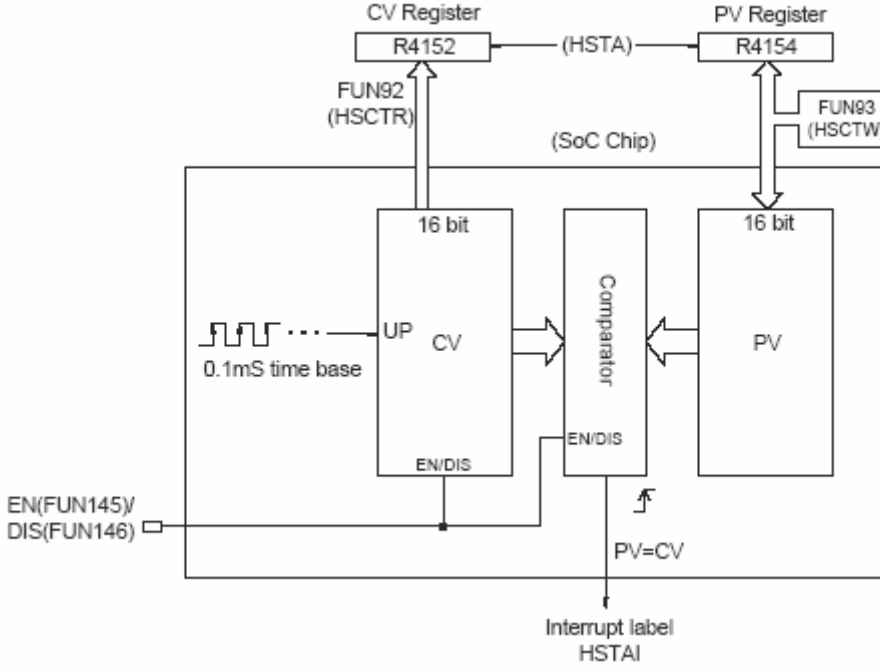
### 10.6.1 HSTA Yüksek – Hızlı Zamanlayıcı

HSTA SoC çipinde yerleşik 16 bitlik bir donanım zamanlayıcısıdır. HHSC gibi, CV okunması için FUN 92 (HSCTR) komutu ve çipteki HSTA PV' ye PV yüklemek için FUN93 (HSCTW) komutu kullanılmalıdır. HSTA iki farklı fonksiyona sahip bir zamanlayıcı gibi de kullanılabilir. FBs-PLC, PV=0 olduğunda 32-bitlik bir döngüsel zamanlayıcı ve PV  $\geq 2$  olduğunda 16-bitlik gecikmeli bir zamanlayıcı gibi kullanacaktır.

## A. HSTA 16-bit yüksek-hız gecikmeli zamanlayıcı

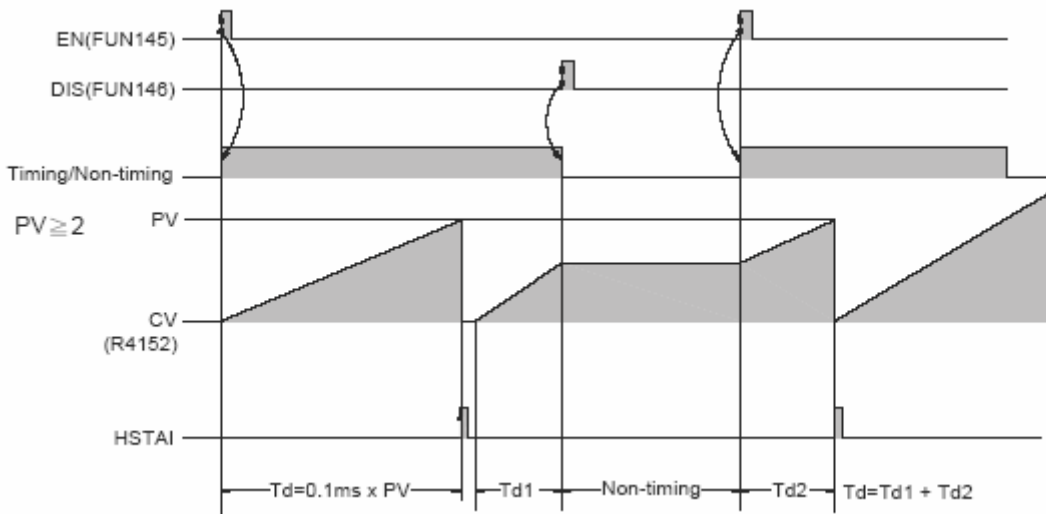
(Uygun interrupt zamanlayıcı)

HSTA zamanlamaya başladıktan sonra, gecikme zamanı bir interruptı çıktısı göndermeden önce PV  $\times 0.1\text{ms}$ ' lik bir gecikme yapacaktır. PV > 0 olduğunda, HSTA 16-bit ve PV 0.2ms~6.5535sn gibi ayarlanabilen bir gecikmeli zamanlayıcı gibi çalışabilir. Mesela; gecikme zamanı 0.2ms~6.5535sn gibi ayarlanabilir. Daha kesin bir zaman tabanına sahip olma ve daha yüksek zaman kesinliği sağlamak için zamanlamaya sürekli dışarıya interrupt gönderebilme dışında, HSTA uygulamaları sıradan bir gecikme zamanlayıcı ile aynıdır. Aşağıdaki diyagram, gecikme zamanlayıcı olarak kullanılan HSTA için diyagram yapısıdır. Uygulama ve fonksiyon detayları için bölüm 10.6.3 "Program Örnekleri" bölümüne bakınız.



SoC çipi içindeki HSTA yüksek-hızlı zamanlayıcısının önceden belirlenmiş register içinde PV yi yazmak için FUN93 uygular. Uygun zamanlı interrupt zamanlayıcı olarak işlem yapacaktır. (Zamanlayıcının her ayar noktası için HSTAI etiketiyle bir kere zamanlı interrupt alt programını yürütecektir.)

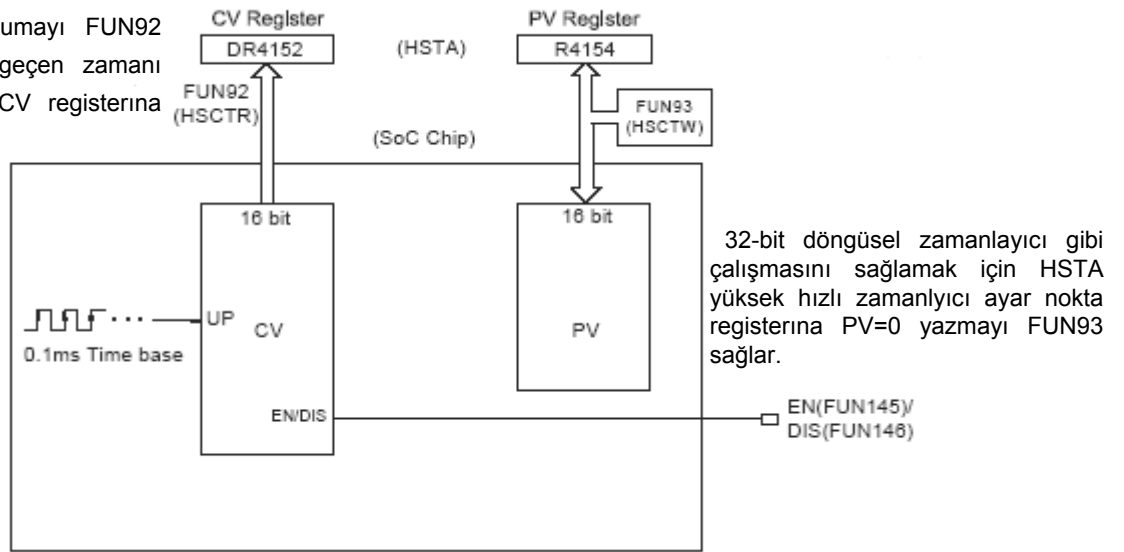
(  
H  
S  
T  
A  
)



## B. HSTA 32-bit Yüksek-Hızlı DÖngüsel Zamanlayıcı

"Döngüsel zamanlayıcı" şeklinde bahsedilen, her sabit bir mesafe için mevcut değere 1 ekleyen ve döngüsel bir şekilde sürekli yukarı sayan bir zamanlayıcıdır. CV değeri 0,1, ...,2147483647,21474836488,2147483649,....429467295,0,1,2..... şeklinde döngüsel olacaktır (zaman tabanı 0.1 ms olduğunda CV değeri, X 0.1 ms biriken süre olacaktır). Gerçekte, döngüsel zamanlayıcı söz konusu iki olayın meydana gelmeleri aralığında geçen süreyi hesaplamak ve meydana geldikleri sırada iki olayı kaydetmek için kullanılmış ve sınırsız bir şekilde çalışabildiği 1ms'lik bir zaman tabanına sahip yukarı doğru sayan döngüsel zamanlama saatidir. Aşağıdaki gösterilen B diyagramı 32-bit döngüsel zamanlayıcı gibi kullanılmış olan HSTA yapısıdır. Bu diyagramda gösterildiği gibi döngüsel zamanlayıcı PV=0 olduğunda, interrupt gönderilmeyecektir. Zamanlama değerini elde etmek için, SoC çipinden CV değerini erişmek için FUN92 kullanmak ve PLC'deki 32-bit registera onu kaydetmek gerekir. Döngüsel zamanlayıcının tipik uygulaması, dönme hızındaki değişim çok büyük veya çok küçük olduğu durumlar altında dönme hızının algılanması daha hassastır. Açıklama için Bölüm 10.6.3'teki örneğe bakınız.

SoC çipindeki HSTA yüksek hızlı sayıcının CV'sini okumayı FUN92 sağlar ve kullanıcı geçen zamanı bilmek için sonucu CV registerına depolar.



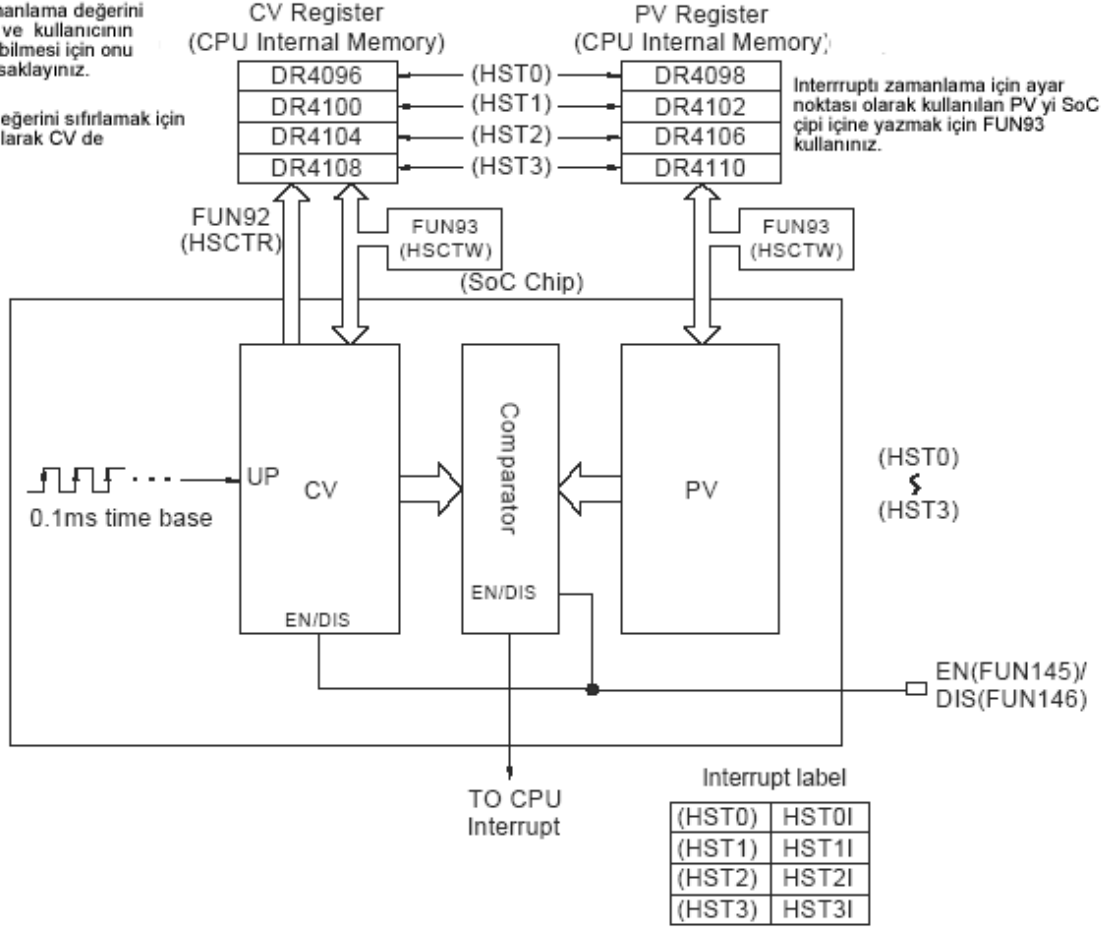
## 10.6.2 HST0~HST3 Yüksek-Hız Gecikmeli Zamanlayıcı

### A. HST0~HST3 Yüksek Hız Gecikmeli Zamanlayıcı (Uygun zamanlı interrupt zamanlayıcı)

HHSC (HST0 ~ HST3), dört adet 32 bit yüksek-hızlı gecikmeli zamanlayıcı olarak yapılandırılabilir. Onlar 16 bit HSTA gecikme zamanlayıcısı ile aynı özelliklere sahiptirler. HST0 ~ HST3, HHSC yi planlamak için 32 bit iken; HST nin, HSC/HST/INT nin Öğe 8'i altında HSC/HST öge seçimi içinde 1 in seçilmesinin yeterlidir. Lütfen Bölüm 11,4 "HSC/HST Yapılandırması" içindeki örneğe (HSTC1, HST! Olarak yapılandırılmak) bakınız. Aşağıdaki çizelge HST olarak planlanan HHSC için fonksiyon yapısının diyagramıdır. Uygulamaları 16 bit HSTA nın işlemleriyle aynıdır. Lütfen Bölüm 11.6.4 "Program Örnekleri" ne bakınız.

SoC çipi içindeki mevcut zamanlama değerini okumak için FUN92 kullanınız ve kullanıcının mevcut zamanlama değerini bilmesi için onu CPU içindeki CV registerına saklayınız.

Aynı zamanda zamanlama değerini sıfırlamak için SoC çipi içine FUN93 kullanılarak CV de yazılabilir



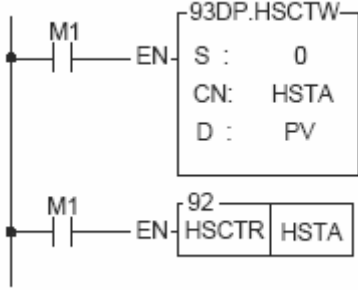
### B. HST0~HST3 32-bit Dönüşsel Zamanlayıcı

Taleb doğrultusunda, HHSC (HST0 ~ HST3), HST0 ~ HST3 ün 32 bit zamanlayıcısı olarak yapılandırılmıştır. Her 0.1 ms' lik aralık için SoC çipindeki mevcut zamanlama değer registerı 1 arttırmış olacaktır. Kullanıcılar mevcut zamanlama değerini okumak ve onu CPU CV registerlarında (DR4096,DR4100,DR4104 ve DR4108) saklamak için FUN92 komutunu kullanabilirler.Bu yüzden 0,1,2,.....7FFFFFFFH,8000000H,.....FFFFFFH,0,1,.....varsayılan değerlerinde olurlar. Olaylar arasındaki aralığı saymak için kullanılan zamanlama hesabı tekniğiyle, 0,1 ms 32-bit zamanlayıcıların sonsuz numarası elde edilebilir.



### 10.6.3 Yüksek –Hızlı Zamanlayıcı HSTA Uygulaması İçin Örnekler

#### Örnek1 32 bit cycle timeri olarak HSTA

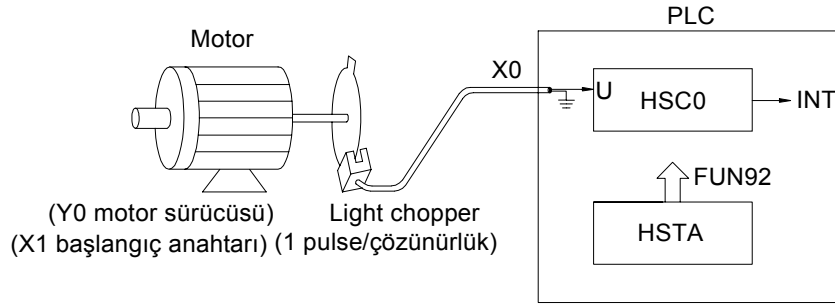


- FUN93 SoC çipinde HSTA PV içine ayar değerini yazmayı sağlar.  
CN =4 olduğunda HSTA gösterir.  
D =1 olduğunda PV gösterir.
- FUN92 SoC çipinde HSTA'nın güncel zamanlama değerini okumayı ve DR4152' ye depolamayı sağlar. (DR4152 değeri 0,1,2 ,.FFFFFF,0,1,2.....dönüşsel varyasyonunda değişir,birim 0.1 ms' tir.)
- CN =4 olduğunda HSTA' yı gösterir

#### Örnek 2 Döngüsel Zamanlayıcı için Örnek Uygulama

Bu örnek, HSTA yı bir dönüşsel zamanlayıcı olarak kullanır. HSC0 ile birlikte çalıştırarak, 10 pulse' in toplaması için zaman aralığını okumak ve 10 pulse'in toplandığı her zaman interrupt yollamak ve karşılıklı olarak gereken RPM i bulmak için kullanılır.

#### Mekanizma

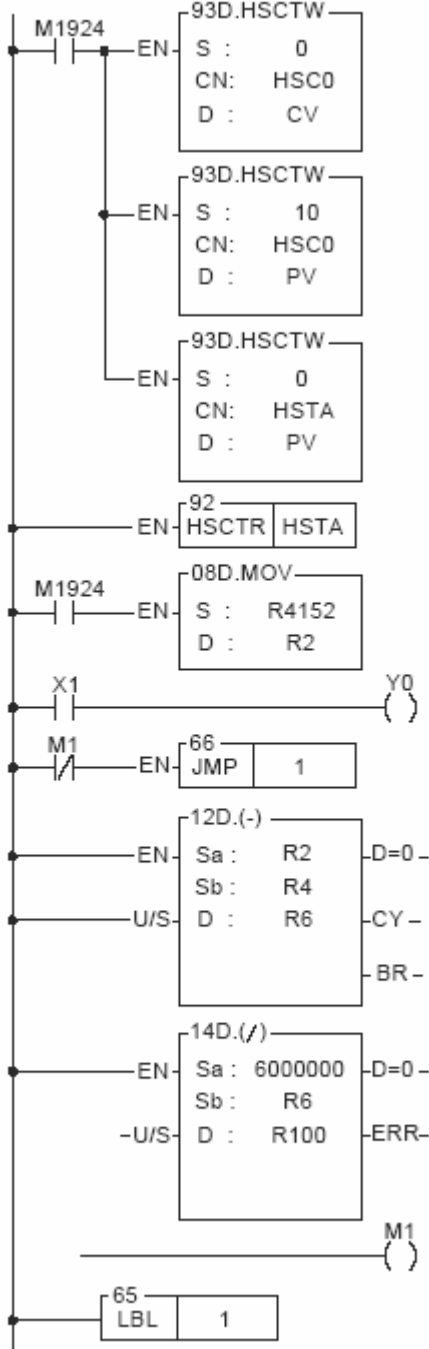


#### HSC ve HST yapılandırması

1. HSTA yerleşiktir ve yapılandırma gerektirmez. Basit olarak PV yi 32 bit dönüşsel zamanlayıcı olarak işlem görmesi için PV=0 yapılır.
2. Foto interrupt ile birlikte çalışmak için,HSC0 ı,tek girişli yukarı sayma sayıcı olarak ayarlayın.(MD0,fakat sadece U girişi kullanın.)

\* Tüm diğer ayarlar (sayımın çokluğu ve kontrol girişi) önceden belirlenmiştir ve değişemez.

【Ana Program】



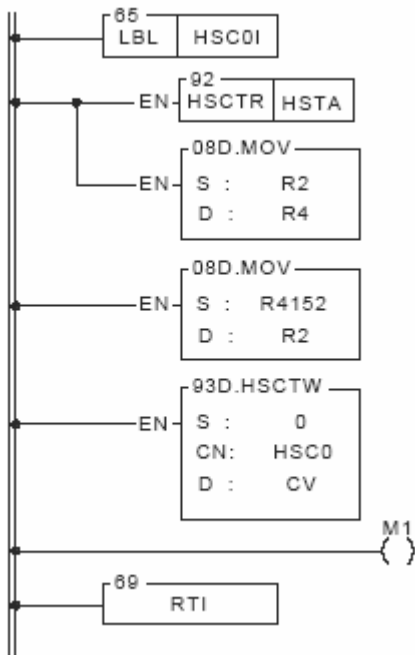
- Mevcut değeri SoC çipi içindeki HSC0 ın CV si içine yazmak için FUN93 kullanınız.  
CN =0, HSC0 yi temsil eder.  
D =0, CV i temsil eder.
- SoC çip içindeki önceden belirlenmiş registera 10 yazar.  
FUN93 CN=0 HSC0 gösterir ve D=1 PV gösterir.
- Önceden belirlenmiş register içine 0 yazar ve HSTA 32-bitlik yüksek hızlı döngüsel zamanlayıcı gibi yapılandırılmıştır.  
FUN93 CN=4 HSTA' yı gösterir ve D=1 PV' yi gösterir.

- Güncel zamanlama değerini okur.
- HSTA CV registerının başlangıç değerini DR2 içinde depolanmıştır.
- Her HSC0 interrupt için aralık bulur,
- Dönüş hızı 
$$= \frac{N}{\Delta T} \times 60 \text{ RPM}$$

$$N=10, \Delta T = \Delta CV \times 0.1 \text{ms} = \frac{(\text{currentCV} - \text{previousCV})}{10000 \text{S}}$$

- Bu yüzden dönüş hızı 
$$= \frac{6000000}{\Delta CV} \text{ RPM}$$
- RPM hesaplama bayrağını temizleyin.

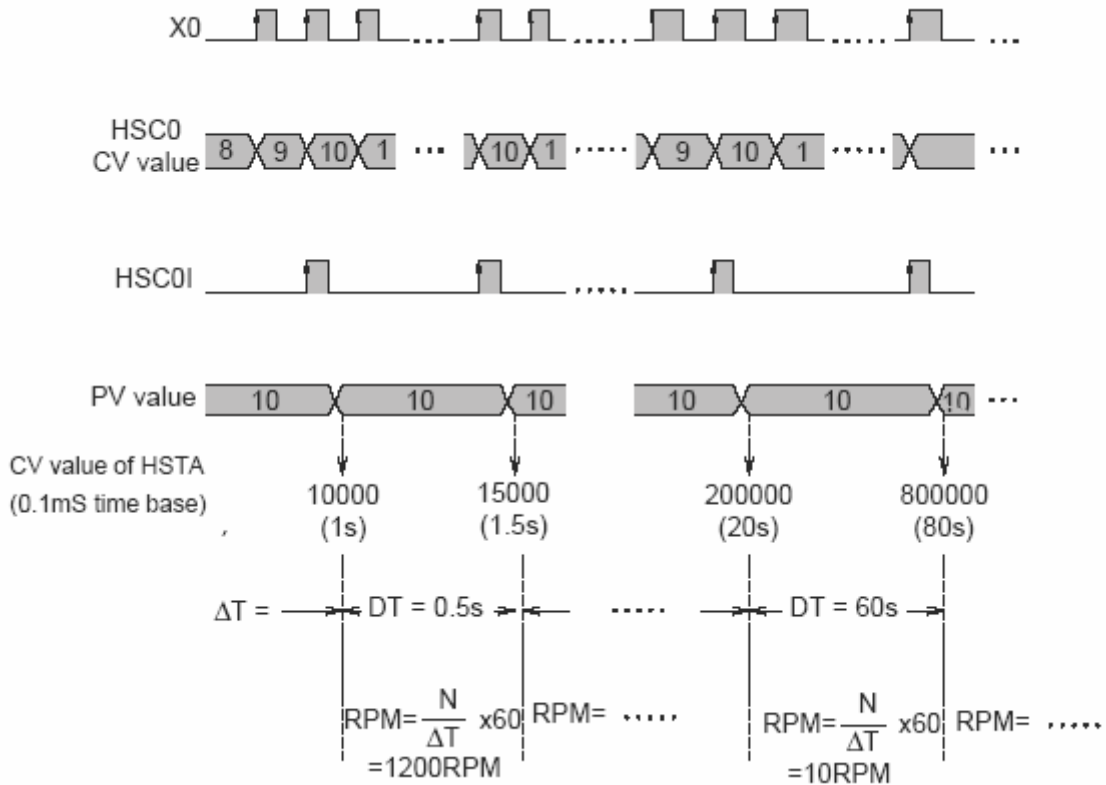
**【Alt Program】**



- Her seferinde, HSC0 10 pulse biriktirdiğinde donanım bu interrupt programını otomatik olarak çalıştıracaktır
- HSTA CV yi okur.

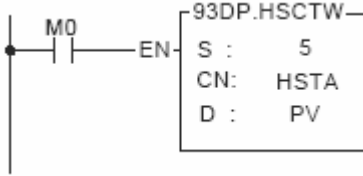
- Mevcut değeri sıfırlar.

- M1=ON, RPM hesaplama bayrağı



### Örnek 3 HSTA uygun zamanlı interrupt zamanlayıcı programı gibi çalışması

#### 【Ana Program】

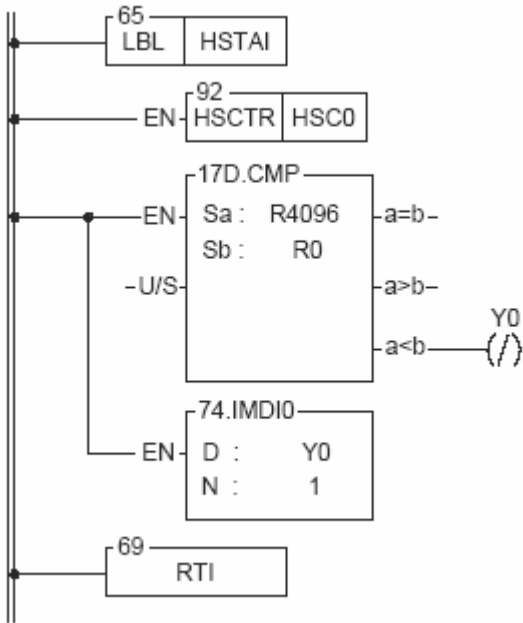


- Uygun zamanlı interrupt zaman periyodunu ayarlar. PV=5 olduğunda, her 0.5ms de HSTAI etiket ismi ile interrupt servis alt programı gerçekleştiğini gösterir.
- SoC çipinde HSTA PV içine ayar değerini yazmayı FUN93 sağlar. Interrupt ayar değeri için time up gibi çalışır.

CN =4, HSTA yı temsil eder.

D =1, PV 'i temsil eder.

#### 【Alt program】

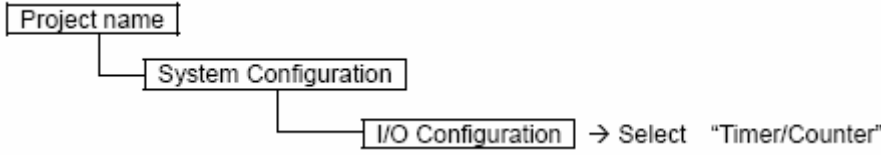


- HSTA etiket isimli interrupt servisi alt programı
- Her 0,5 mS' de bir kez HSC0 donanım yüksek hızlı sayıcının güncel değerini okur.
- Güncel sayma değerinin R0' dan büyük mü yoksa eşit mi olduğunu gösterir. Eğer evet ise Y0 ON olacaktır.
- Yüksek-hızlı çıkış tepkisine ulaşmak için Y0 çıkışını hemen günceler (Aksi takdirde tarama zamanında gecikme olacaktır)

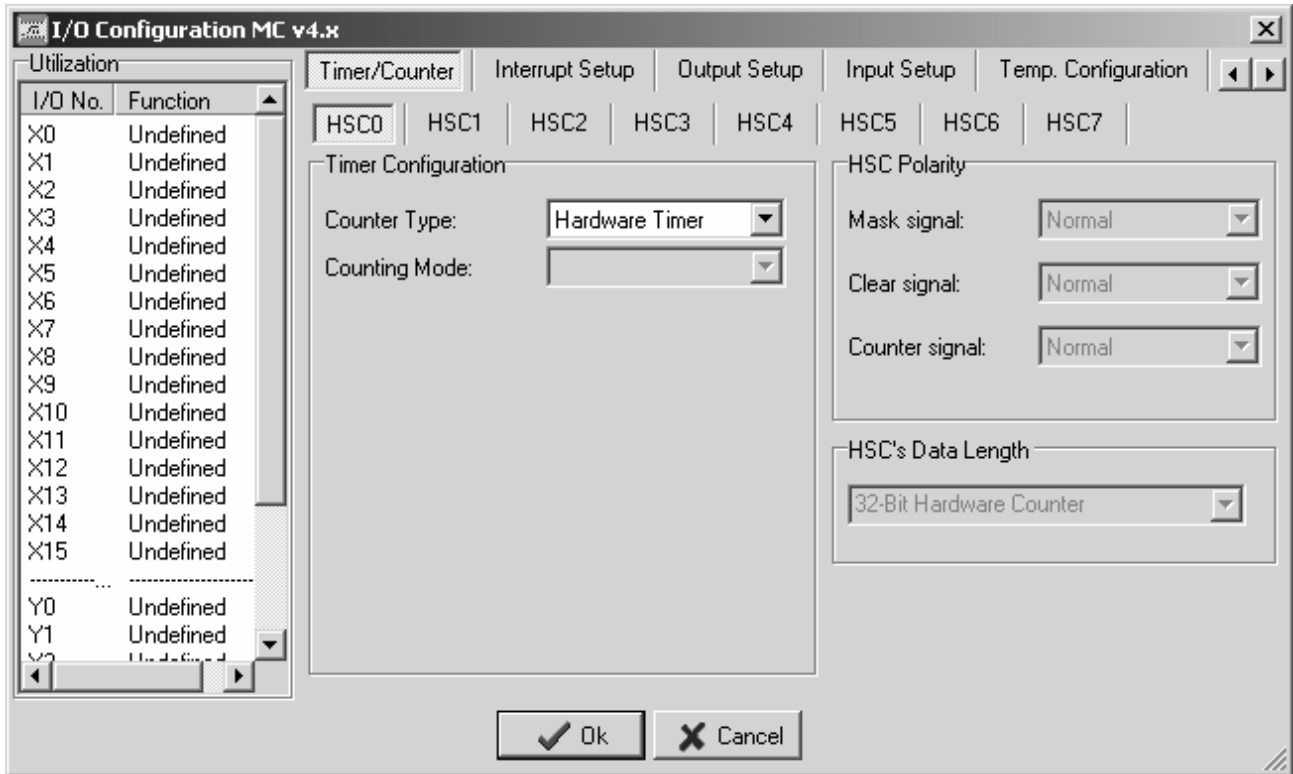
## 10.6.4 Yüksek-hızlı Zamanlayıcı HST0~HST3 Uygulama Örnekleri

### HSC ve HST yapılandırması (WinProLadder kullanarak)

Proje penceresindeki "I/O Configuration" kısmına tıklayın :



- "Zamanlayıcı/Sayıcı" penceresi görüldüğüne, sayıcı tipi olarak "Hardware Timer" seçebilirsiniz, bundan sonrada HHSC (Donanım yüksek hızlı sayıcı) HHT (Donanım Yüksek Hızlı Zamanlayıcı) atanmaları için yapılandırılabilir.
- Kullanıcıların HSTA' yı yapılandırmasına gerek yoktur, çünkü HSTA defaulttur. Sadece HHSC' nin (Yüksek-Hızlı Donanım Sayacı) HHT (Yüksek-Hızlı Donanım Zamanlayıcı) olması istendiğinde yapılandırılması gerekir.



### HSC ve HST yapılandırması(FP-07C kullanarak)

```
HSC0/HST0 SELECT
1 (0:HSC, 1:HST)
```

```
HSC1/HST SELECT
0 (0:HSC, 1:HST)
```

```
HSC1→ MD:0 U:X4
D: M: C:
```

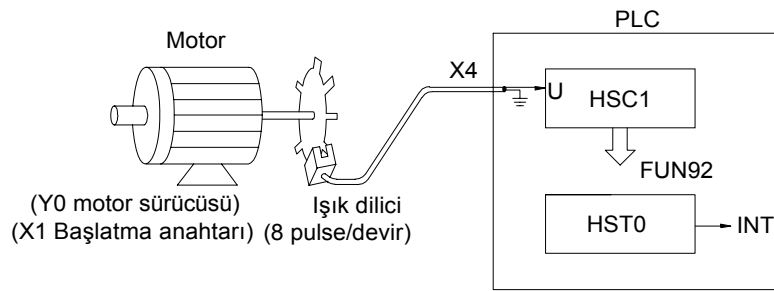
- HSC, HST0 olarak ayarlandı
- HSC1, HSC gibi belirlenmiştir
- HSC1, tek bir girişle bir yukarı sayma sayıcısı MD0 olarak ayarlanmıştır. Diğer girişler kullanılmaz

- Tüm diğer ayarlar (sayımın çoğulluğu ve kontrol girişi gibi)defaulttur ve değiştirilmemelidir.

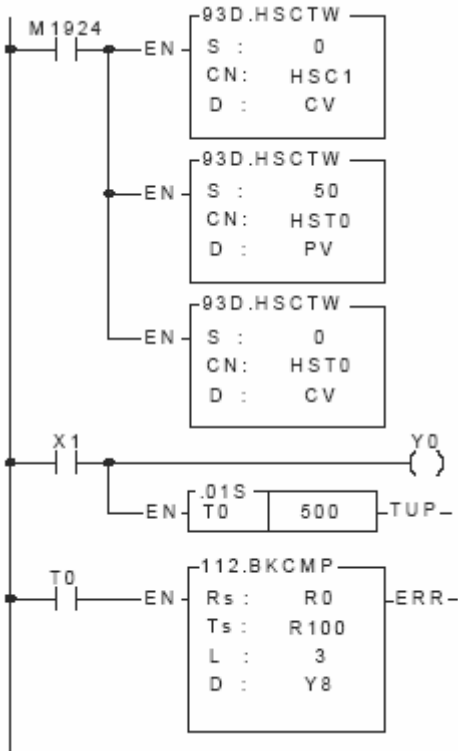
## Örnek 1 Gecikme Zamanı için Uygulama Örneği

Bu örnekte HSC0, HST0 gecikme zamanlayıcısı olarak yapılandırılır. Aynı zamanda, bir otomatik odun delme makinesinin döner motoru ile yüksek-hızlı sayıcı HST1 bağlanır ve sabit periyotta çıkışa bir interrupt gönderir. Her interrupt oluştuğunda sayıcının saydığı değer okunacaktır. Sonra, yükleme olmadığında ve delme başlığı indirildiği zamanki motor rotasyon sayısı arasındaki hız değişimi karşılaştırılarak motor RPM değişimi hesaplanabilir. Motor RPM'i delme başlığı normal olduğunda, kör olmasından daha hızlı olucak ve daha az direnç gösterildiği farz edilmiştir. Matkap başı kırıldığında, direnç olmaz ve RPM çok yüksek olduğundan bir delme işlemi yokmuş gibi çalışır. Üç durum arasında dönme hızındaki fark önemsizdir. Bu farkı saptaması 10ms'den fazla olan herhangi bir zamanlayıcı algılayamaz ve örnekleyemez. Yine de, interruptla birleştirilmesi 0,1 ms' lik zaman zaman aralığına sahip HST uygulanmış, matkap başının durumu (normal, kör veya kırık) fark edilmiş olabilir. Bu sebepten matkap başı değişimi için uyarı verilebilir veya işlem durdurulabilir (zaman sabittir ve darbe sayısı değişkendir).

## Mekanizma

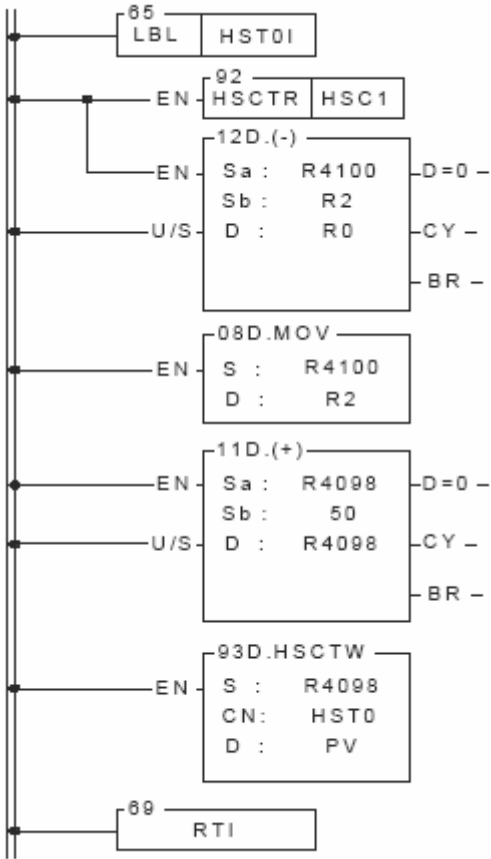


## 【Ana Program】



- FUN93 SoC çipindeki güncel değer registerını resetlemeyi sağlar. CN=1,HSC1'yi gösterir ve D=0, CV yi gösterir.
- HST0 PV 50 olarak ayarlanmıştır, örneğin; her 0.5 ms de bir interrupt gönderir (50 x 0.1ms)
- HST0 CV registerının başlangıç değeri 0' dır.
- 5 saniye motor çalıştıktan sonra kesici başlığın RPM hızını karşılaştırmak için FUN112'yi kullanır  
R0: Her 5 mS de elde edilen HSC1 darbe sayısı

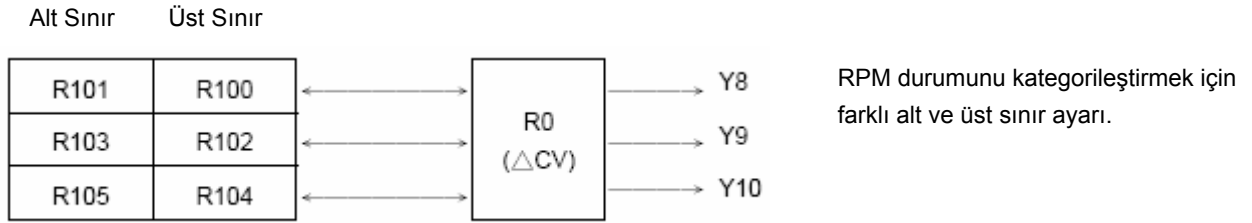
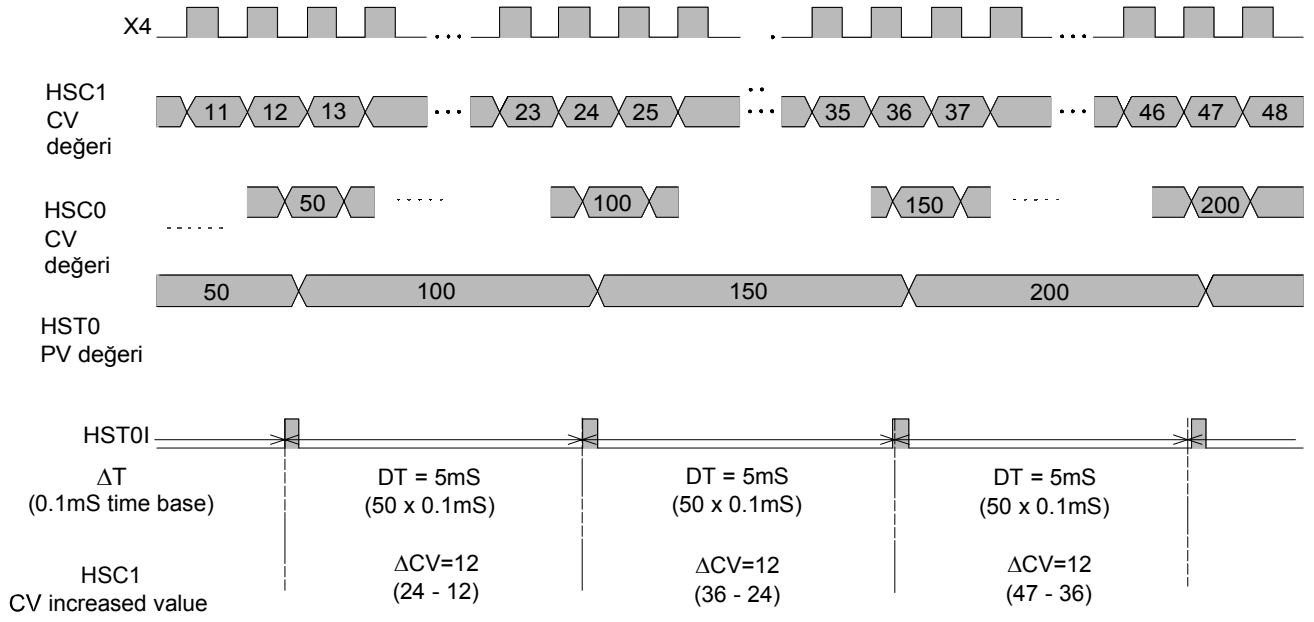
### 【Alt Program】



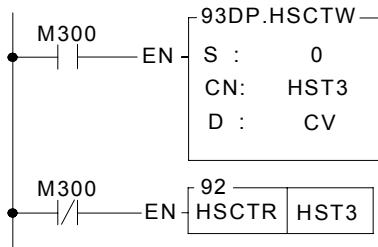
- Donanım her 5 ms de bir kez bu alt programı çalıştıracaktır.
- HSC1' in güncel sayma değerini okur ve onu DR4100 içine koyar.
- Bu 5ms aralığındaki HSC1 CV artmasını bulup ve onu DR0 içine depolar
- Yeni HSC0 PV' yi hesaplar

### 【Açıklama】

Matkap başının normal RPM i 18000rpm sayıldığında, işli interrupt bir devirde 8 pulse üretecektir. Bu durumda HSC1' in U pininin frekansı  $18000/60 \times 8=2400\text{Hz}$  dir. Örneğin; her 5 ms için 12 darbe üretilecektir. Bu yüzden, HST0 bir interrupt göndermek için kullanılabilir ve RPM değerini elde etmek için her 5 ms HSC1 CV değerini okuyabilir.



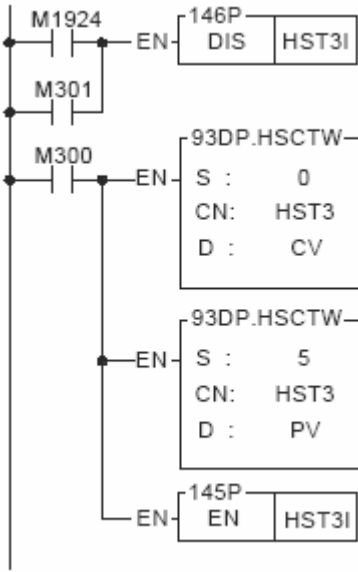
**Örnek 2 32-bit döngüsel zamanlayıcı gibi çalışan Donanımsal Yüksek Hızlı Zamanlayıcı HST3**



- M300 0 → 1 değiştiğinde, Güncel değer registerı 0 olur.
- FUN93, SoC çipinde HST3 CV içine güncel değere 0 yazmayı sağlar. CN = 3 olduğunda HST3 gösterilir. D = 0 olduğunda CV gösterilir.
- FUN92, SoC çipinde HST3'ün güncel zamanlama değerini okumayı sağlar ve güncel değer registerı DR4108 içine depolar. (DR4108 değeri 0, 1, 2, ....., FFFFFFFF, 0, 1, 2, ..... şeklinde döngüsel olarak değişir. Birim 0.1mS dir.) CN = 3 olduğunda HST3 gösterilir.

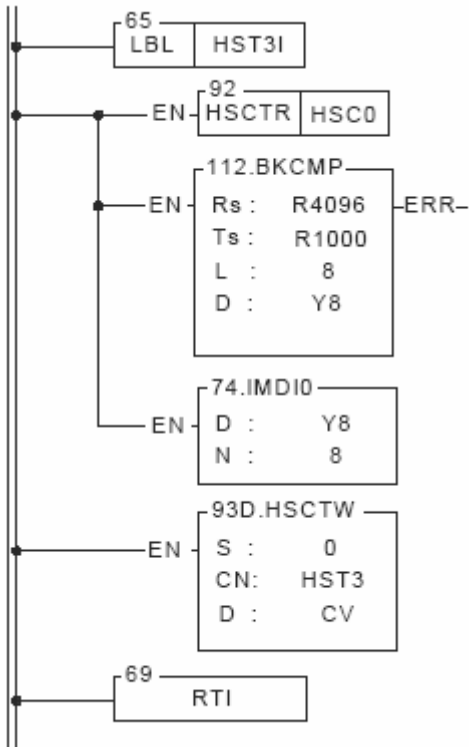


**【Ana Program】**



- Konum değiştiğinde veya M301 ON olunca, HST3 ün periyodik interrupt göndermesine engel olacaktır.
- M300 0→1 olduğunda, mevcut registerı 0 yapar.
- FUN93, SoC çipinde HST3 CV içine güncel değer olarak 0 yazmayı sağlar.  
CN =3 olduğunda HST3 gösterir; D=0 olduğunda CV gösterir.
- Periyodik interrupt aralığını ayarlar; PV=5 olunca HST3I etiket isimli interrupt servis altprogramını her 0.5ms de bir kez sağlayacağını gösterir.
- FUN93, SoC çipindeki HST3 PV içine ayar değerini yazmayı sağlar. Interrupt ayar değeri time up gibi davranır.  
CN=3 olduğunda HST3 gösterir; D=1 olduğunda PV gösterir.
- HST3 interruptı enabledir.

**【Alt Program】**



- HST3I etiket isimli donanım yüksek hızlı interrupt servisi alt programı.
- Her 0,5 ms de donanım yüksek hızlı sayıcının güncel değerini okur.
- Elektronik tamburun bölgesi güncel sayma değer düşümünü gösterir ve buna denk çıkış noktasını ON yapar.
- Çıkış Y8~Y15 aralığını hemen güncelleyiniz.
- FUN93, SoC çipinde HST3 CV içine güncel değer registerını resetlemeyi sağlar.  
CN=3 olduğunda HST3 gösterir; D=0 olduunda CV gösterir.



# KISA NOTLAR