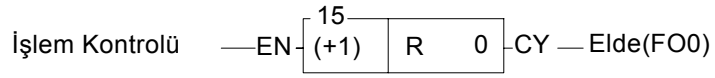
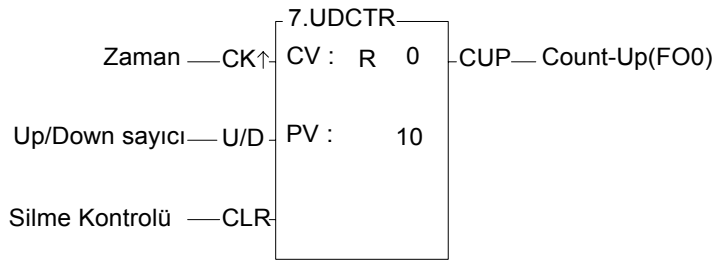
Bölüm 5 Fonksiyon Komutlarının Tanımları

5.1 Fonksiyon Komutlarının Formatı

Bu bölümde FBs-PLC' nin fonksiyon komutlarını detaylı olarak anlatacağız. Her bir fonksiyon , tüm açıklamalar giriş kontrolü, numara/isim komutu, operand ve fonksiyon çıkışı dahil dört parçaya bölünecektir. Eğer FB–07 kullanımı giriş mnemonik komutuna ise, T, C, SET, RST and SFC komutları hariç FP-07'de tek bir tuşa basarak direk olarak girilebilir, diğer fonksiyon komutlarından daha çok komut numarası tuşlanarak girilmelidir. Altta bir örnek gösterilmiştir.

Ladder Diyagram	FP–07 Mnemonic Kod
Örnek 1: Tek Giriş Komutu	
	FUN 15 D: R 0
Örnek 2: Çoklu Giriş Komutu	
	FUN 7 CV: R 0 PV: 10

Mnemonik kod alanındaki boş kutu içindeki wordler D:, CV:, ve Pr: gibi FP-07'den yapılan mesajlaşma kullanıcı tarafından girilemez.

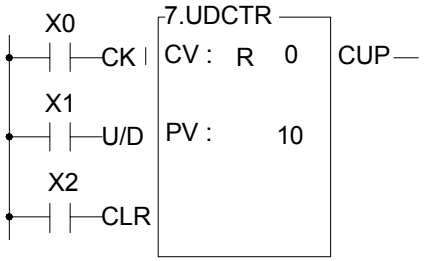
5.1.1 Giriş Kontrolü

Yedi fonksiyon komutu hariç giriş komutu yoktur. Diğer FBs-PLC fonksiyon komutlarının giriş kontrol numarası birden dörde kadardır. Komut ve işlemlerin çalışması, giriş kontrol sinyaline veya çeşitli giriş kontrol sinyallerinin kombinasyonlarına bağlıdır. Fatek yazılımı Winprollader karmaşık tasarımları tamamlayarak kullanıcıya yardımcı olmaktadır. Ladder program penceresindeki fonksiyon komutlarının tümüne bakınız, içerdiği girişler, çıktılar, fonksiyon ismi ve parametre isimleri kısaltılmış wordlere çevrilmiş bloklar halinde gösterilmiştir. Üstte örnek 2'de gösterildiği gibi, "CK↑" girişi sayıcı 0'dan 1'e (yükselme kenarı) değişip 1 artıp ya da azaldığında ilk giriş "CK↑" gösterimi işaretlenecektir. ("U/D" durumuna göre). İkinci giriş işareti "U/D", ("U") aralığının üstündeki wordde sunulanlar 1'in durumu ve ("D") aralığının altındaki wordde sunulanlar o'ın durumudur. ikinci giriş "U/D" durumu=1 sayıcı,"CK↑" girişi O'dan 1' e değiştiğinde 1 artacak ve "U/D"=o olduğunda 1 azalacaktır. Üçüncü giriş işareti "CLR" göstergeleri giriş 1 olduğunda sayıcı silinecektir. Bölüm 8~9'da her bir fonksiyon komutunun giriş kontrolünün tanımları verilmiştir.

Açıklama: Başlama hattına direk olarak bağlanan yedi komut MCE, SKPE, LBL, RTS, RTI, FOR ve NEXT' dir. Daha detaylı bilgi için bölüm 6 ve 7'ye bakınız.

Fonksiyon komutlarının tüm giriş kontrolleri uygun elemanlar tarafından bağlanmalıdır aksi takdirde bir söz dizimi hatası oluşacaktır. Altta örnek 3'de gösterildiği gibi, FUN7 fonksiyon komutu, FUN7 önce üç girişe ve üç elemana sahiptir. ORG X0, LD X1 ve LD X2 ilk giriş CK↑, ikinci giriş U/D ve üçüncü giriş CLR ile uyuşurlar.

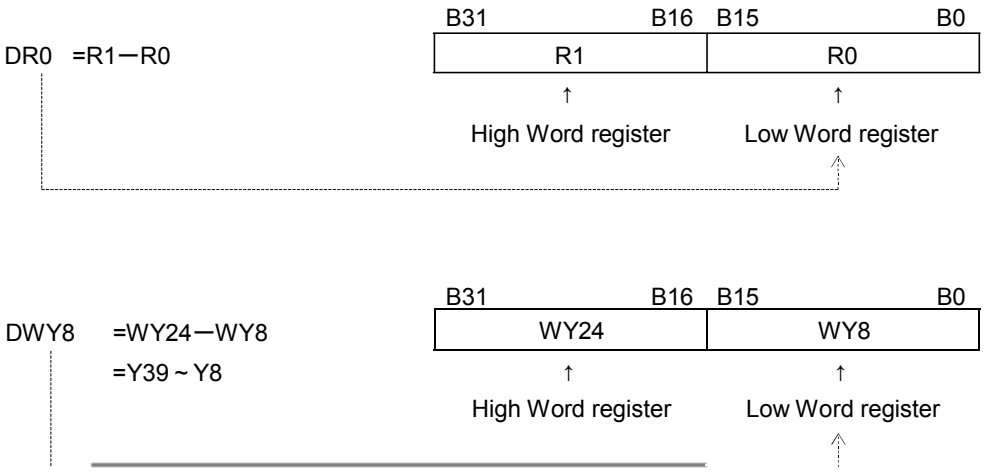
Örnek 3:

Ladder Diyagram	FP–07 Mnemonic Kod
	ORG X 0 LD X 1 LD X 2 FUN 7 CV : R 0 PV : 10 FUN7 üç eleman gerekmektedir. çünkü 3 girişi vardır

5.1.2 Komut Numarası ve Türev Komutları

Önceden bahsedildiği gibi, dokuz komut haricinde klavyede atanmış tuşlar kullanılarak girilebilir, diğer fonksiyon komutları ise "komut numarası" kullanılarak girilmelidir. Takip eden üç komut numarası sonlarına D,P,DP eklenmiş ve ek olarak bu üç fonksiyon komutu türemiştir.

D: Bir Double wordu gösterir (32-bit). 16-Bit word FBs-PLC' deki kayıtların temel ünitesidir. R, T ve C (C200~C255 hariç) kayıtlarının data uzunluğu 16-bittir. Eğer 32-bit uzunluğundaki kayıtlar gerekli ise sonra R1-R0, R3-R2 v.b. 2 ardışık 16-bit kayıtlarla birlikte birleştirilmesi gerekir. Register isimlerinden önce D harfli bir ön ek kullanılarak DR0'da R1-R0 ve DR2'de ise R3-R2 için kullanılır. Eğer FP-07'nin izleme modundaki DR0 veya DWT8 girişiniz ise, sonrasında 32-bit uzunluğunda bir değer (R1-R0 veya WY24-WY8) gösterilecektir.

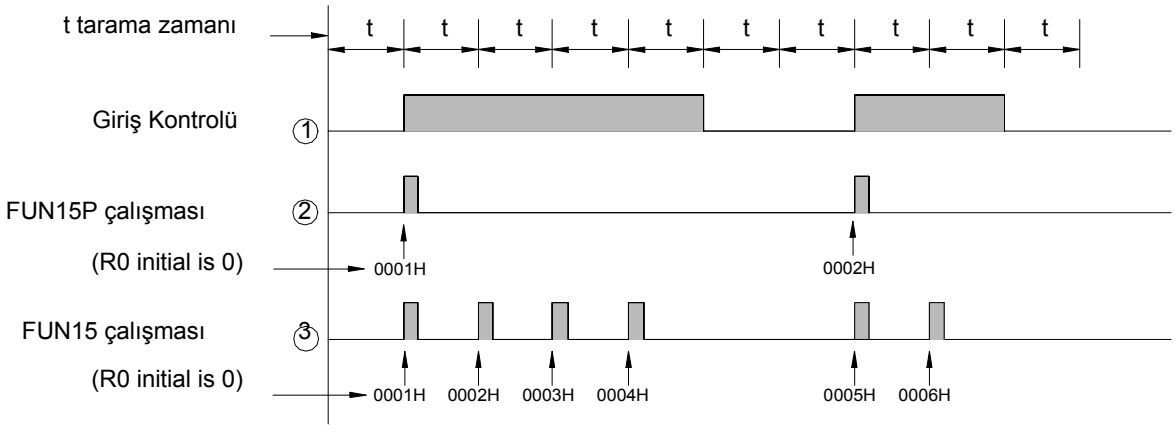


Açıklama: Diyagram ve Mnemonik kod kullanılırken 16-bit ve 32-bit komutları arasındaki farkın ayırt edilmesi için, 32-bitlik komutların "komut numarası"' ndan önce ön ek olarak D harfi eklenir ve diğer uygulananın boyutu örnek 4'te gösterildiği gibi 32-bit olur. FUN 11D komutu ön ek olarak D harfine sahiptir. Bu yüzden, kaynak ve hedef uygulananları D harfli bir ön eke ihtiyaç duyarlar. Sa: RO toplanacak sayı gibi Sa=DR0=R1-R0 ve Sb=DR2=R3-R2. Kaynak ve hedef hariç diğer operandların uzunluğunun , 1 word 16 bit veya 32 bitlik komutlar ile kullanılmasına özellikle dikkat edilmelidir.

P: Darbe modu komutunu gösterir. Giriş kontrolünün durumu 0'dan 1'e (yükselen kenar) değiştiğinde Komut çalışacaktır. Örnek 1'de gösterildiği gibi, eğer ön ek P harfi komuta eklenmiş ise (FUN 15P), Giriş kontrol sinyalinin durumu 0'dan 1'e değiştiğinde FUN 15P komutu çalışacaktır. Eğer bir P ön ekine sahip değilse komutun çalışması seviye modudur. Bu sayede komut 1'den 0'a değişen giriş kontrolünün durumuna kadar sürekli tarama için çalışacaktır. Darbe girişi "↑", CK↑, EN↑, TG↑ vb. semboller ile gösterilmiştir. Bu işlem kullanımında, bir fonksiyon komutunun işlem ifadesinin örneği altta gösterilmiştir.

- İşlem kontrolünde〝EN〞=1 or〝EN↑〞(P komutu) 0→ 1, ……

İlk önce P olmayan komut için (seviye modu) çalışma gereksinimleri gösterir ve ikinci olarak da P komutlu (darbe modu) çalışma gereksinimlerini gösterir. Aynı giriş durumlarında FUN15 ve FUN15P'nin sonuçları (R0) aşağıdaki dalga formunda gösterilmiştir.



DP: Komut, darbe modlu 32-bit bir komutu göstermektedir.

Açıklama: P komutu, program taramaktaki seviye komutundan daha fazla zaman harcar, bu yüzden kullanıcı mümkün olduğu kadar P komutunu kullanmalıdır.

5.1.3 Operand

Operand data kaynağı ve depolanması için kullanılır. Kaynak (S) uygulananın datası sadece kaynak içindir ve komutun çalışması ile değişecektir. Hedef (D) uygulanan işlem sonucu depolamakta kullanılır ve data komutun çalışmasından sonra değişebilir. Aşağıdaki tabloda gösterilen FACON PLC fonksiyon komutlarının operandleri, registerlar, bobinler ve kontakların tipleri operand gibi kullanılabilirler.

■ Özel Operandlerin İsimleri ve Fonksiyonları:

Kısaltma	İsim	Tanımlar
S	Kaynak	Kaynak (S) operandinin datası sadece kaynak ve okumak içindir ve komutun çalışması ile değişmeyecektir. Eğer kaynak operandı birden fazla ise her bir operand Sa ve Sb gibi dipnotlar ile tanımlanır.
D	Hedef	Hedef (D) operandin işlem sonucunu depolamakta kullanılır. Orjinal data işlem sonrasında değişebilecektir. Sadece bobinler ve registerlarin yazmaya korumalı olmayanları hedef operand olabilirler.
L	Uzunluk	Tablonun uzunluğu veya boyutu gösterimleri genellikle sabittir.
N	Sayı	Bu sabitler çoğunlukla numaralar ve zamanlardır. Eğer sabit birden fazla ise, her bir sabit Na,Nb,Ns... vb. gibi dipnotlar tarafından tanımlanacaklarıdır.
Pr	İşaretleyici	Tablodaki kaydı veya spesifik bir datayı veya spesifik bir data bloğunun işaretlemekte kullanılır. Genellikle Pr değeri değiştirilebilmektedir. Bu yüzden giriş kayıtları ve sabit olamazlar (R3840~R3847).
CV	Anlık Değer	T ve C komutlarındaki T veya C'nin anlık değerini depolamakta kullanılırlar.
PV	Kurma Değeri	Karşılaştırma ve kaynak için T ve C komutlarında kullanılır
T	Tablo	Ardışık kayıt formları bulunan bir tablonun kurulma kombinasyonudur. Temel işlem üniteleri tek ve double worddür. Eğer bir tablodan fazla iseler, her bir tablo Ta, Tb, Ts and Td v.b. gibi dipnotlar ile tanımlanmışlardır.
M	Matrix	Ardışık kayıt formları bulunan bir matrisin kurulma kombinasyonudur. Temel işlem ünitesi bittir. Eğer bir matrisden fazla iseler, her bir matris için Ma, Mb, Ms and Md v.b. gibi dipnotlar ile tanımlanacaklardır.

Üstte bahsedilen özel operandler haricinde, diğer operandler frekans için Fr, yığın için ST, kuyruk için QU v.b. gibi belli özel amaçlar için kullanılırlar. Daha fazla detay için komut tanımlarına bakınız.

■ Operand ve diğer aralık Tipleri: Fonksiyon komutu için operand tipleri ayrık, register ve sabitlerdir.

a) Ayrık operand :

Toplamda beş fonksiyon komutu vardır ve SET, RST, DIFU, DIFD ve TOGG gibi adlandırılmışlardır. Bu beş komut sadece Y□□□ (harici çıkış), M□□□□ (dahili ve özel) ve S□□□ (step) rölelerin işlemleri için kullanılırlar. Beş fonksiyon komutunun aralıkları ve gösterimleri aşağıdaki tabloda gösterilmektedirler.

Range / Ope-rand	Y	M	SM	S
	Y0 | Y255	M0 | M1911	M1912 | M2001	S0 | S999
D	○	○	○ *	○

D (hedef uygulanan) gösterilen "O" sembolü uygulananlar gibi bobinlerin bu tipinde kullanılırlar. SM sütununda gösterilen "O" üzerindeki "*" işareti operandler gibi yazımı yasaklanmış röleler hariç tutulmalıdır.

Özel röle komutları için sayfa 12-8'e bakınız.

b) Register operand :

Fonksiyon komutları için ana operand register operandidir. Register operandlerinin iki tipi vardır: negatif registerlar R, D, T, C gibi wordler veya double wordlerdir. Diğeri türev registerlarıdır (WX, WY, WM, WS) ve bunlar ayrık bit formundadırlar. Registerların tipleri komut operandleri gibi kullanabilmektedirler ve aralıkları aşağıdaki tabloda gösterilmişlerdir:

Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
Ope-rand	WX0 | WX240	WY0 | WY240	WM0 | WM1896	WS0 | WS984	T0 | T255	C0 | C255	R0 | R3839	R3840 | R3903	R3904 | R3967	R3968 | R4167	R5000 | R8071	D0 | D4095	16/32-bit +/- number	V、Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○ *	○	○	○
D		○	○	○	○	○	○		○	○ *	○ *	○		○
•														

Tabloda gösterilen "o" sembolü operand gibi datanın bu çeşidine de uygulanabilmektedir. "○*" sembolü ile gösterilenler operand gibi yazımı yasaklanmış registerler hariç diğer dataya uygulanabilir. Yazımı yasaklanmış registerler ve özel register komutları hakkında daha fazla için sayfa 3-6'ya bakınız.

R5000 ~ R8071 sadece okunabilir registerlara set edilemezler, normal registerlar gibi kullanılabilirler (okuma ve yazma).

Açıklama 1: WX, WY, WM and WS gibi W ön ekli registerlar 16 bit formundadır. Örneğin, WX0 register vasıtasıyla X0(bit 0)~X15(bit 15) formundadır. WY144 register vasıtasıyla Y144(bit 0)~Y159(bit 15) formundadır. Ayrık numaraların 0, 8, 16, 24... gibi 8'in katları olmasına dikkat edin.

Açıklama 2: Tablodaki son register (word) fonksiyondaki 32-bitlik bir operand gibi sunulmamaktadır. çünkü 2 word 32- bitlik operand için gereklidir.

Açıklama 3: TMR (T0 ~ T255) ve CTR (C0 ~ C255) sırasıyla zamanlayıcı ve sayıcı registerleridir. Her ne kadar genel registerlar gibi kullanılsa da onlar aynı zamanda sistemleri güçleştirip daha zor hata ayıklamaları yapmaktadırlar. Bu yüzden TMR veya CTR registerlar içine herhangi bir şey yazmaktan kaçılmalıdır.

Açıklama 4: T0 ~ T255 ve C0 ~ C199 16-bitlik registerlardır. Ama C200~C255 32-bitlik registerdır. Bu yüzden 16-bitlik operandler kullanılmazlar.

Açıklama 5: R0 ~ R8071'in aralığındaki registerların operandı dolaylı adresleme yapmak için V veya Z pointer registerleri ile birleştirilebilir. Dolaylı adresleme yapmak için pointer register (XR) kullanımının tanımlanmasına bölüm 5.2'deki örneğe bakınız.

c) Sabit operandler :

16-bit'lik sabit -32768~32767 arasındadır. 32-bitlik sabit -2147483648~2147483647 arasındadır. Çeşitli fonksiyon komutları için sabit sadece pozitif sabit olabilmektedir. 16-bit ve 32-bit sabitin aralığı aşağıdaki tabloda listelenmiştir.

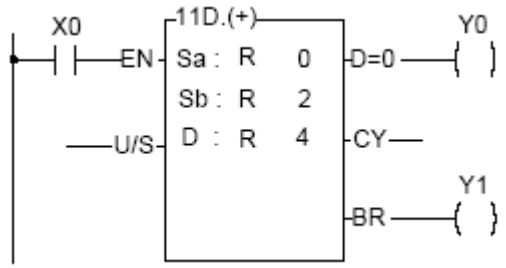
Sınıflandırma	Aralık
16-bit işaretlenmiş sayı	-32768 ~ 32767
16-bit işaretsiz sayı	0 ~ 32767
32-bit işaretlenmiş sayı	-2147483648 ~ 2147483647
32-bit işaretsiz sayı	0 ~ 2147483647
16/32-bit işaretlenmiş sayı	-32768 ~ 32767 or -2147483648 ~ 2147483647
16/32-bit işaretsiz sayı	0 ~ 32767 or 0 ~ 2147483647

L, bit boyutu, N v.b. gibi spesifik bir operandin uzunluğu ve boyutu farklıdır ve farklılıklar operand sütunundakilerin tümüne direk olarak etiketlenmiştir. Fonksiyon komutlarının açıklamasına bakınız.

5.1.4 Fonksiyon Çıkışı (FO)

"Fonksiyon Çıkışı" (FO) fonksiyon komutunun işlem sonucunu göstermekte kullanılır. Kontrol girişi gibi, programlama yazılımının ekranındaki her bir fonksiyon çıkışının kısaltmalarının tümü bir wordle bağlanmıştır ve çıkış fonksiyonel olarak kısaltmalardan oluşmaktadır. Carry den türetilmiş olan CY gibi. Çıkış fonksiyonunun maksimum sayısı 4'tür ve bunlar FO0, FO1, FO2, FO3 şeklinde sembolize edilmişlerdir. FO durumu Fo komutundan çıkarılmaktadır (FP-07 programını yazan vihazda FO özel tuşu vardır). Kullanılmamış FO herhangi bir elemana bağlı olmadan bırakılabilir. Örnek 4'de gösterildiği gibi.

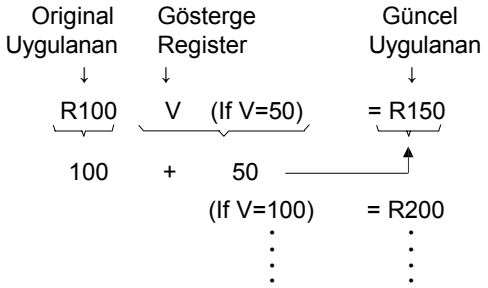
Örnek 4 :

Ladder Diyagram	Mnemonic Kodlar
	ORG X 0 FUN 11D Sa: R 0 Sb: R 2 D : R 4 FO 0 OUT Y 0 FO 2 OUT Y 1

M1919=0 olduğunda, eğer komut çalışıyorsa FO durumları güncellenecektir. Komut tekrar çalıştıktan sonra, yeni BirFO olusana kadar aynı durumda kalacaktır (hafızada tutularak).
M1919=1 olduğunda, komut çalışmıyorsa FO durumları resetlenecektir (hafızada tutulmayacak).

5.2 Dolaylı Adresleme için Index Regisdteri (XR) Kullanımı

FBs-PLC fonksiyon komutlarındaki, bazı operandleri dolaylı adresleme yapmak için (V、Z、P0~P9) pointer registerları ile kombine edilebilir (eğer uygulanabilirse operand tablosunda gösterilecektir). Ancak, sadece R0～R8071 aralığındaki registerlar dolaylı adresleme gerçekleştirmek için pointer registeri ile birleştirilebilir. (diğer operandler ayrık gibidir, sabit ve D0～D3071 dolaylı adresleme için kullanılamazlar).
On iki gösterge registeri vardır XR (V、Z、P0~P9). V registeri gerçekte özel register R4164'dir. Z registeri R4165 ve P0~P9 registeri (D4080~D4089)'dir. İndex adreslemesi tarafından güncel adreslenmiş register, index registerinin içeriği ile orjinal uygulanan sapacaktır.



Alttaki diyagramda gösterildiği gibi, operand adresini değiştirmek için sadece V değerini değiştirmek gerekir. FBs-PLC fonksiyon komutu ile index adresleme birleşimi sonrasında güçlü ve yüksek derecede bir verim kontrolü çok basit bir şekilde sağlanabilir. Örnekte alttaki diyagramda program kullanılarak, dinamik blok data displayi sağlamak için bir data taşıma komutuna ihtiyaç duyulmuştur, mesela sistem yönetim molası gibi.

Index Register(P0~P9) Komutu

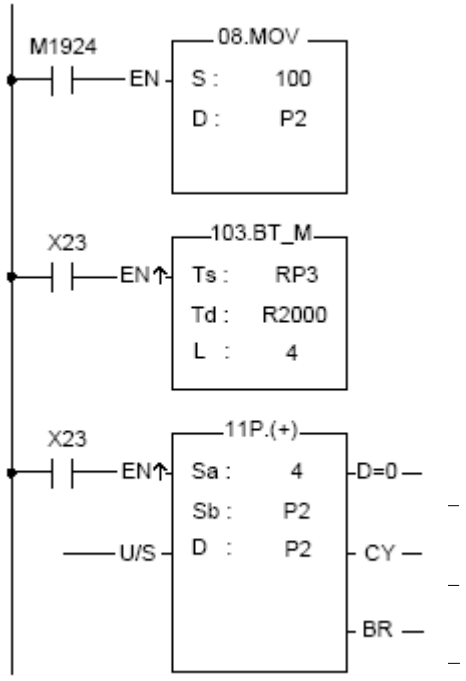
Dolaylı adresleme uygulamasında, Rxxxx registeri index adresleme için V, Z & P0~P9 birleştirilmelidir; Dxxxx registeri index adresleme için V, Z ile kombine edilemez, ama P0~P9 izin verilmiştir.

V, Z index registeri Rxxxx register ile birleştirildiğinde, örneğin V, Z'li R0, komut formu R0V(Burada V=100, R100) veya R0Z(Burada Z=500, R500) şeklindedir.

P0~P9 registeri Rxxxx registeri ile birleştirildiğinde komut formu RPn (n=0~9) veya RPmPn (m,n=0~9) olur, örneğin RP5 (burada P5=100, R100) veya RP0P1(burada P0= 100, P1=50, 150).

P0~P9 index registeri Dxxxx registeri ile birleştirildiğinde komut formu DPn (n=0~9) veya DPmPn(m,n=0~9) biçimindedir. Örneğin P3 (burada P3=10, D10) veya DP4P5 (burada P4=100, P5=1, it means D101).

P0~P9 index registerının her ikisi de kullanılabilir, örneğin P2=20, P3=30. Rxxxx or Dxxxx registerının her ikiside index registeri ile birleştiğinde RP2P3, R50'yi gösterecektir. Bunun yolu dolaylı adresleme için her iki index registerinin toplanmasıdır.



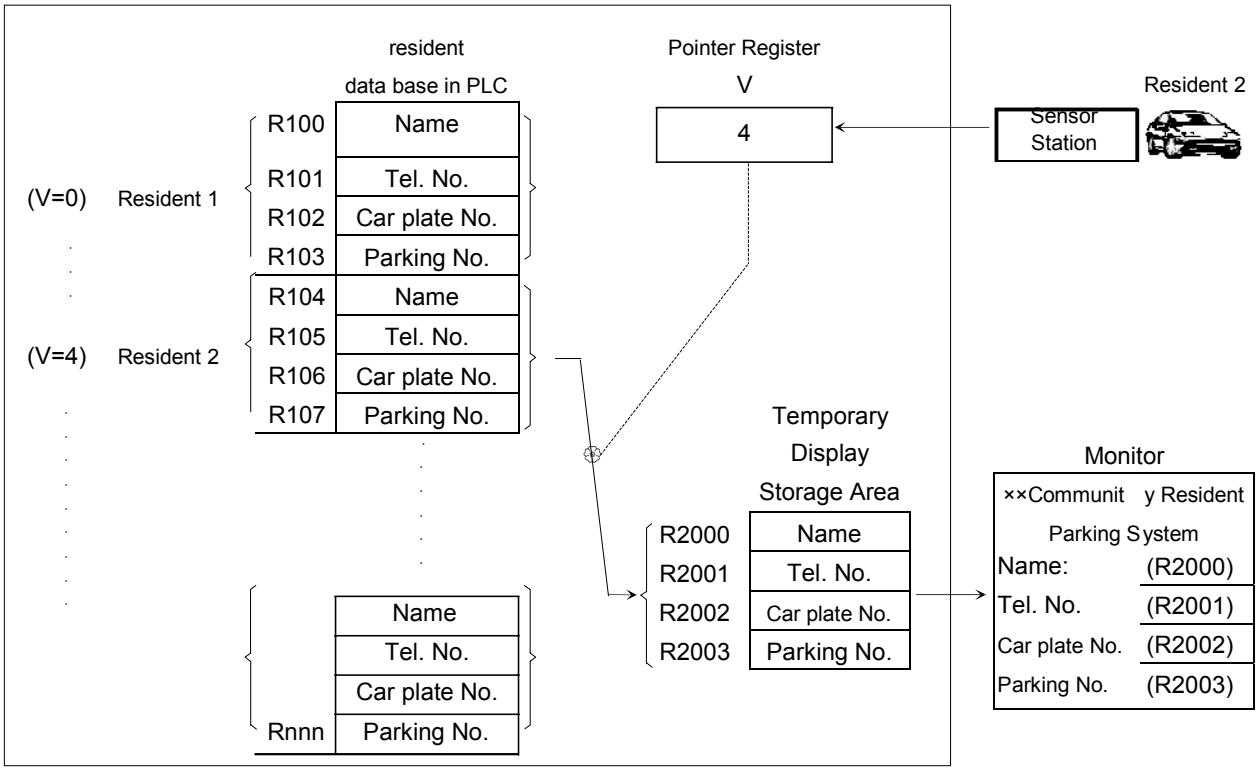
1. Güç verilirken veya ilk çalışmada index registeri P2=100'dir.

2. X23 0 ⇨ 1 değiştiğinde, FUN103 tablo hareketini sağlayacaktır, kaynak R100 (P2=100)'den başlar hedef R2000'den başlar ve miktarı 4'tür.
   ilk çalışmada R2000~R2003 için R100~R103'in içeriğini kopyalar, ikinci çalışmada R2000~R2003 için R104~R107'in içeriğini kopyalar.

3. P2 index registeri 4 ilerdeki noktaya 4 arttırılır.

Dolaylı Adresleme Program Örneği

Ladder Diyagram	Mnemonic Kodlar
103.BT_M EN Ts : R100 V Td : R2000 L : 4	ORG SHORT FUN 103 Ts : R100V Td : R2000 L : 4



Tanımlama Topluluk yerleşimleri için mola yönetim sistemindeki yerleşik park alanının 100 olduğu varsayılmaktadır. Her bir yerleşim, levha ve park numarası, telefon numarası ve isim dahil temel bilgiler yerleştirilmiştir. Bu aşağıdaki şekilde gösterildiği gibi dört ardışık PLC registerını kapsamaktadır. Toplam 400 register (R100 ~ R499) ayrılmıştır. Her bir yerleşime park öbeği veya ana girişin hassas girişi için eşsiz kart numaralı (numara yerleşik 1 için 0, yerleşik 2 için 4 v.b.) bir kart verilmiştir. Kart numarası PLC tarafından algılanacak ve "V" pointer registerina depolanacaktır. Sonucun izlenmesi PLC'deki R2001 ~ R2003 tarafından kavranılan datayı görüntüleyecektir. Örneğin, 4 kart numaralı 2 yerleşimin kartı algılanmıştır. sonra register V=4 ve PLC geçici display depolama alanına (R2000 ~ R2003) R104 ~ R107'deki data hemen taşınacaktır. Ancak, sonucun izlenimi kartın algılamasının hemen ardından yerleşik 2'nin datasını görüntüleyecektir.

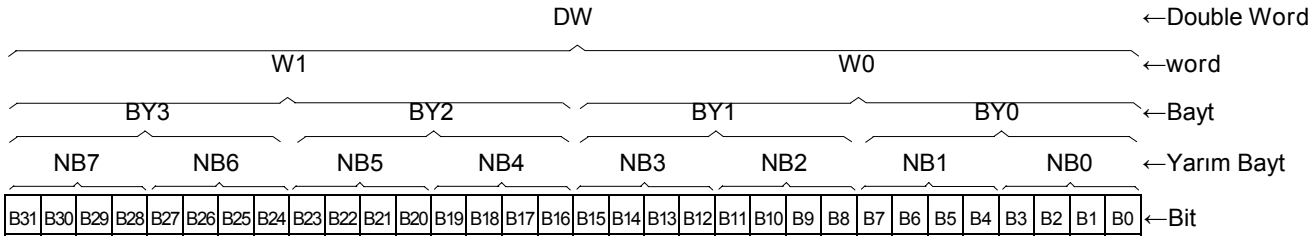
□ Uyarı
1. Dolaylı adresleme için pointer registeri kullanılmasına rağmen uygulama güçlü ve esnektir, ama V ve Z değerleri serbestçe ve dikkatsizce seçilerek normal data alanlarına yanlış yazılmasıyla büyük hasarlar oluşabilir. Kullanıcı işlem esnasında özel önlem almalıdır. 2. Data register aralığındaki R3840 ~ R4167 (mesela IR, OR veya SR) da bulunan 328 register dolaylı adresleme uygulaması için kullanılabilen sistem ve I/O kullanımı için ayrılmış önemli registerlardır. İsteğe bağlı olarak bu registerlar yazılarak önemli sistem veya I/O hataları oluşabilir ve önemli sorunlara sebep olabilir. Gerçekten kullanıcılar, V ve Z değerleri tarafından esnek bir şekilde register adres değişimlerini kontrol edemez ve kolayca algılayamaz, FBs-PLC, hedef adres R3840 ~ R4067 aralığındaysa otomatik olarak denetleyecektir. Eğer öyleyse, yazma işlemi çalışmayacak ve M1969 bayrağı "Dolaylı adreslemenin illegal yazılması" 1 olacaktır. O durumda R3840 ~ R4067 registerlarına yazmak gerekmektedir. Bunun için direk adreslemeyi kullanın.

5.3 Sistem Numaralandırma

5.3.1 Binary Kod ve İlgili Terminoloji

Binary dijital bilgisayarların numaralandırma sistemidir. Ayrık ON/OFF değerleri ile PLC işlemleri için binary kod kullanmak doğaldır. Aşağıdaki terimler numaralandırma sisteminin daha fazla konuya gitmeden önce tamamıyla anlaşılmalıdır.

- Bit: (B gibi kısaltılmış, B0,B1 ve diğerleri gibi) Binary değerin en temel ünitesidir. Bitlerin durumu "1" veya "0"'dır.
- Yarım Bayt: (NB şeklinde kısaltılmıştır, NB0, NB1 v.b.) ardışık dört bit tarafından biçimlenmiş ve 0 ~ 9 arasında bir decimal sayı veya 0 ~ F arasında bir hexadecimal sayı kullanılmalıdır.
- Bayt: (Kısaltması BY'dir. BY0,BY1 gibi kullanılır). 2 ardışık yarım baytlar ile oluşturulmuştur (8 bit , B7 ~ B0) ve 2 basamaklı hexadecimal sayılar kullanılmıştır.
- Word: (Kısaltması W'dir. W0,W1 gibi kullanılır). 2 ardışık bayt ile oluşturulur (16 bit gibi, B15 ~ B0) ve 4 basamaklı 0000 ~ FFFF hexadecimal sayılar kullanılır.
- Double Word: (Kısaltması DW'dir. DW0,DW1 gibi kullanılır). 2 ardışık word (32 bit, B31 ~ B0 gibi) ile oluşturulur ve 8 basamaklı hexadecimal sayı kullanır.



- Ondalık Sayılar :

IEEE-754 standartını takip eden Fatek-PLC'nin ondalık sayı formatının depolanması için double word kullanılır ve bu aşağıda ifade edilmiştir:

Float tipindeki sayı = işaret + Üst + mantis

Sign	Exponent	Mantissa
b_{22}	b_{30} ~ b_{23}	b_{22} ~ b_0
1 bit	8 bits	23 bits

32 bits

▲ Eğer işaret biti 0 ise sayı pozitiftir. Eğer işaret biti 1 ise sayı negatiftir.

▲ Üstel durum 8 bittir.

▲ Mantisi 2 tabanında 23-bittir. Normalde mantis her zaman 1 bit ile başlar taban noktası tarafından takip edilmiş, mantisin dayanağıdır. 1 bit ilerleyerek normalized bir mantisde daima mevcuttur, örtülüdür ve sunulmaz.

- Tam sayıyı ondalık sayı tipine dönüştürme kuralı :

N = (-1) S * 2 (E -127) * (1.M) 0 < E < 255

Örneğin :

(1) 1 = (-1) 0 * 2 (01111111) * (1.000……0)

İşaret 0 ile temsil edilmektedir, 127 aşımlı üstün kodu 127=01111111'dir ve belirgin bit 1'dir. Mantisdeki sonuçların tümü O olur. 1'in basit duyarlıklı IEEE 754 gösterimi şu şekildedir:

Code(1) =

0	0	1	1	1	1	1	1	1	0	0	0	0	0·····0	0	0
s	e	e	e	e	e	e	e	e	m	m	m	m	m m····· m	m	m

= 3F800000H

(2) 0.5 = (-1) 0 * 2 (01111110) * (1.000……0)

İşaret 0 ile temsil edilmektedir 127 fazlalık üstel kod 127-1=01111110'dir ve belirgin bit 1'dir. mantisteki sonuçların tümü O olur. 0.5'in basit duyarlılıklı IEEE 754 gösterimi şu şekildedir:

Code(0.5) =

0	0	1	1	1	1	1	1	0	0	0	0	0	0----------0	0	0
s	e	e	e	e	e	e	e	e	m	m	m	m	m m····· m	m	m

= 3F000000H

(3) -500.125 = (-1) 1 * 2 (10000111) * (1. 11110100001000000000000)

İşaret 0 ile temsil edilmektedir 127 fazlalık üstel kod 127-1=01111110'dir ve belirgin bit 1'dir. mantisteki sonuçların tümü O olur. -500.125'in basit duyarlılıklı IEEE 754 gösterimi şu şekildedir:

Code(-500.125) =

1	1	0	0	0	0	1	1	1	1	1	1	1	0	1	0	0	0	0	1	0	0	0	00------0	0
s	e	e	e	e	e	e	e	e	m	m	m	m	m	m	m	m	m	m	m	m	m	m	mm·····m	m

= C3FA1000H

5.3.2 FBs-PLC için Numerik Sayıların Kodlanması

FBs-PLC dahili işlemler için binary numaralandırma sistemi kullanır. Harici BCD girişlerinin datası PLC çalışmadan önce binary dataya dönüştürülmelidir. Bilindiği gibi kullanıcılar için PLC 'e binary kod girmek ve okumak çok zordur . Bu yüzden FP-07 ve WinProladder datayı göstermek ve girmek için decimal veya hexadecimal üniteler kullanırlar. Ama gerçekte, tüm işlemler PLC'deki yerleşimlerini binary kod ile gerçekleştirmişlerdir.

Açıklama: Eğer WinProladder veya FP–07 (Örnek için, 7-segment display veya thumb wheel anahtar kullanılarak I/O terminalleri sayesinde çıkış datası alıp veya data içine girebilirler) olmadan datayı gösterebiliyor veya girebiliyorsanız, sonra binary dönüşümü decimal olarak gerçekleştirmek için Ladder program gerçekleştirmelisiniz. Bu seçimleriniz ile WinProladder ve FP-07 kullanılmadan datayı gösterebilir veya girebilirsiniz. FUN20(BIN→BCD) ve FUN21(BCD→BIN) bakınız.

5.3.3 Numerik Değerin Aralığı

Daha önce bahsedildiği gibi FBs- PLC dâhili operasyonlar için binary sayıları kullanır. 16-bit ve 32-bit in FBs- PLc nin 3 farklı nümerik datalarıdır. Bu üç nümerik değer aralığı aşağıda gösterilmiştir.

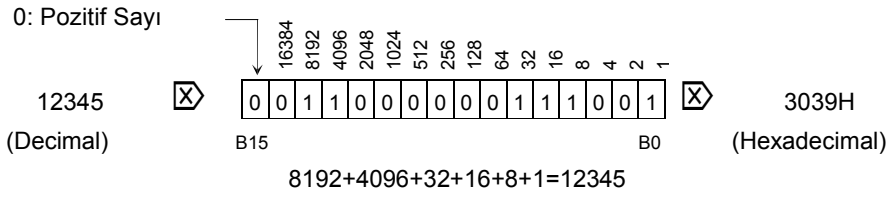
16-bit	−32768 ~ 32767
32-bit	−2147483648 ~ 2147483647
Floating point number	±(1.8*10-38 ~ 3.4*1038)

5.3.4 Numerik Değerin Gösterimi

(Yeni başlıyorsanız bu bölümü geçebilirsiniz)

16-bit ve 32-bit numerik değerlerin gösterimi ve özellikleri, daha karmaşık uygulamalar için numerik değer işlemini kullanıcıya daha rahat anlatma için seçme kısmı altta sağlanmıştır.

16-bit ve 32-bit'in en belirgin bitleri negatif ve pozitif bitleri tanımlamakta kullanılmıştır (0:pozitif ve 1: negatif). Diğer bitler (B14~B0 or B30~B0) sayının büyüklüğünü göstermektedir. Bundan başka açıklama için 16-bit kullanılan örnek aşağıdadır. Bunların hepsi aynı zamanda 32-bitlik sayılara uygulanır ve tek farkı uzunluğudur.

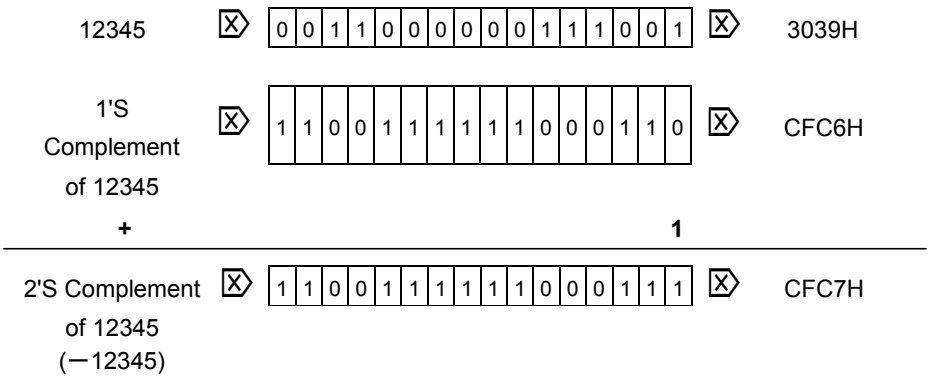


Üstteki örnekte, boyutuna (16-bit veya 32-bit) aldırmadan ve en küçük belirgin LSB biti ile başlar. B0 1, B1 2, B2 4, B3 8 ve bu şekilde devam eder. Soldaki komşu bit tarafından sunulmuş sayı değeri (1, 2, 4, 8, 16 v.b.) double' dır ve değer 1 ile gösterilen bitlerin toplamıdır.

5.3.5 Negatif Sayıların Gösterimi

(Yeni başlayanlar bu bölümü geçmelidir)

MSB 1 olduğunda, sayı negatif sayı olacaktır. FBs-PLC negatif sayılar 2'ye tamamlayanından oluşmaktadır, mesela 1 eklendikten sonra eşdeğer pozitif sayının (1'komplement 1'leri 0 yapar , 0'ları 1 yapar) tüm bitlerini tersler. Üstteki örnekte pozitif sayı 12345'dir. 2'ye tamamlayanın hesaplanması (mesela -12345) altta tanımlanmıştır:



5.4 (+1) Artarak ve (-1) Azalarak Overflow ve Underflow oluşması

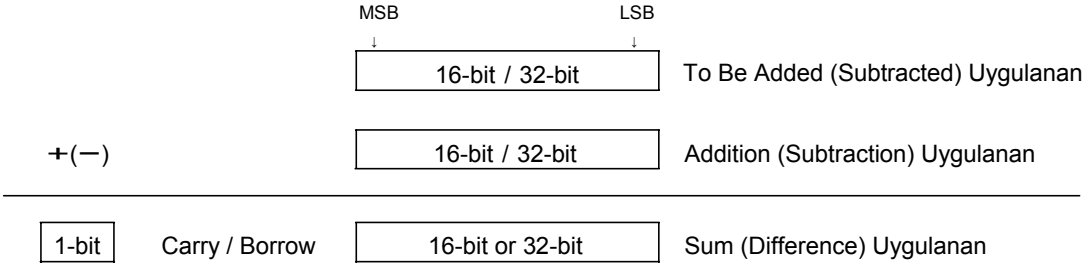
(Yeni başlayanlar bu bölümü geçmelidir)

Maksimum pozitif sayı 16-bit ve 32-bitlik opernadlerde sunulmuştur ve bu değerler sırasıyla 32767 ve 2147483647'dir. Minimum negatif değerler 16-bit ve 32-bit opernadler ile simgelenirken sırasıyla –32768 ve –2147483648 değerlerini alır. Bir operandin artım veya azalımında (mesela; register veya sayıcı değerinin aşağı yukarı değişmesinde +1 veya -1 olur) ve opernadin pozitif limitinin değerini geçen sonuç olduktan sonra "Overflow" (OVF) oluşur. Bu negatif limite dönüş değerini sağlayacaktır (mesela; 16-bit pozitif limitli 32767'ye 1 eklendiğinde –32768 şeklinde değişecektir). Eğer sonuç uygulananın negatif limitinden küçük ise "Underflow" (UDF) oluşur. Bu pozitifi limite döndürecek değer oluşturacaktır (mesela; –32768 negatif değerinden 1 indirirsek 32767 değeri şeklinde değişecektir). overflow veya underflow çıkış bayrağı FO da oluşur. Bu bayrakların 16-bit veya 32-bit işlem sonuçları aşaınını elde etmek için kaskastlanmış komutlar kullanılabilir.

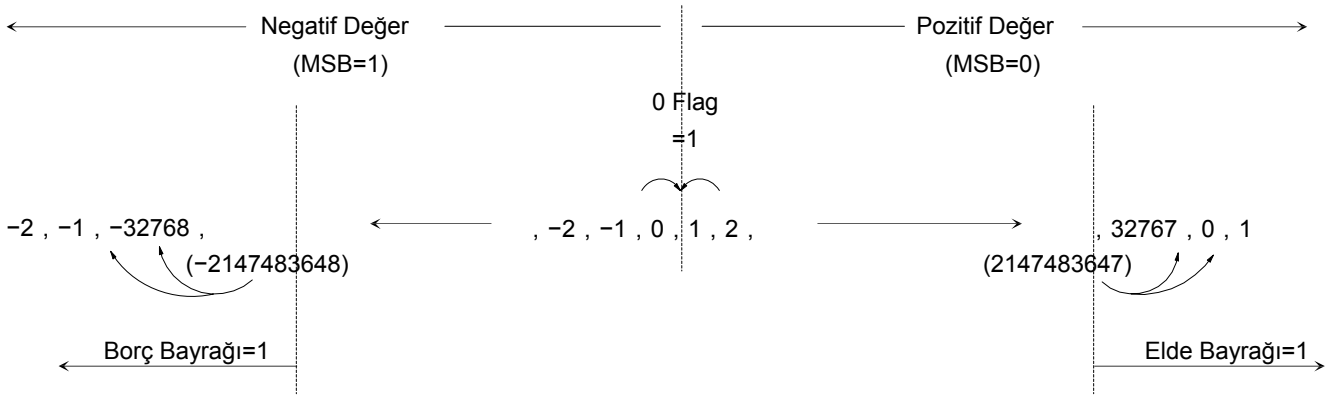
Artım (Azalım) Sonucu / Overflow/ Underflow	16-bit Operand	32-bit Operand
Artma	OVF=1 → −32767 → −32768 → 32767 → 32766 → 32765	OVF=1 → −2147483646 → −2147483647 → −2147483648 → 2147483647 → 2147483646
Düşme	UDF=1 → −32767 → −32768 → 32767 → 32766 → 32765	UDF=1 → −2147483647 → −2147483648 → 2147483647 → 2147483646 → 2147483645

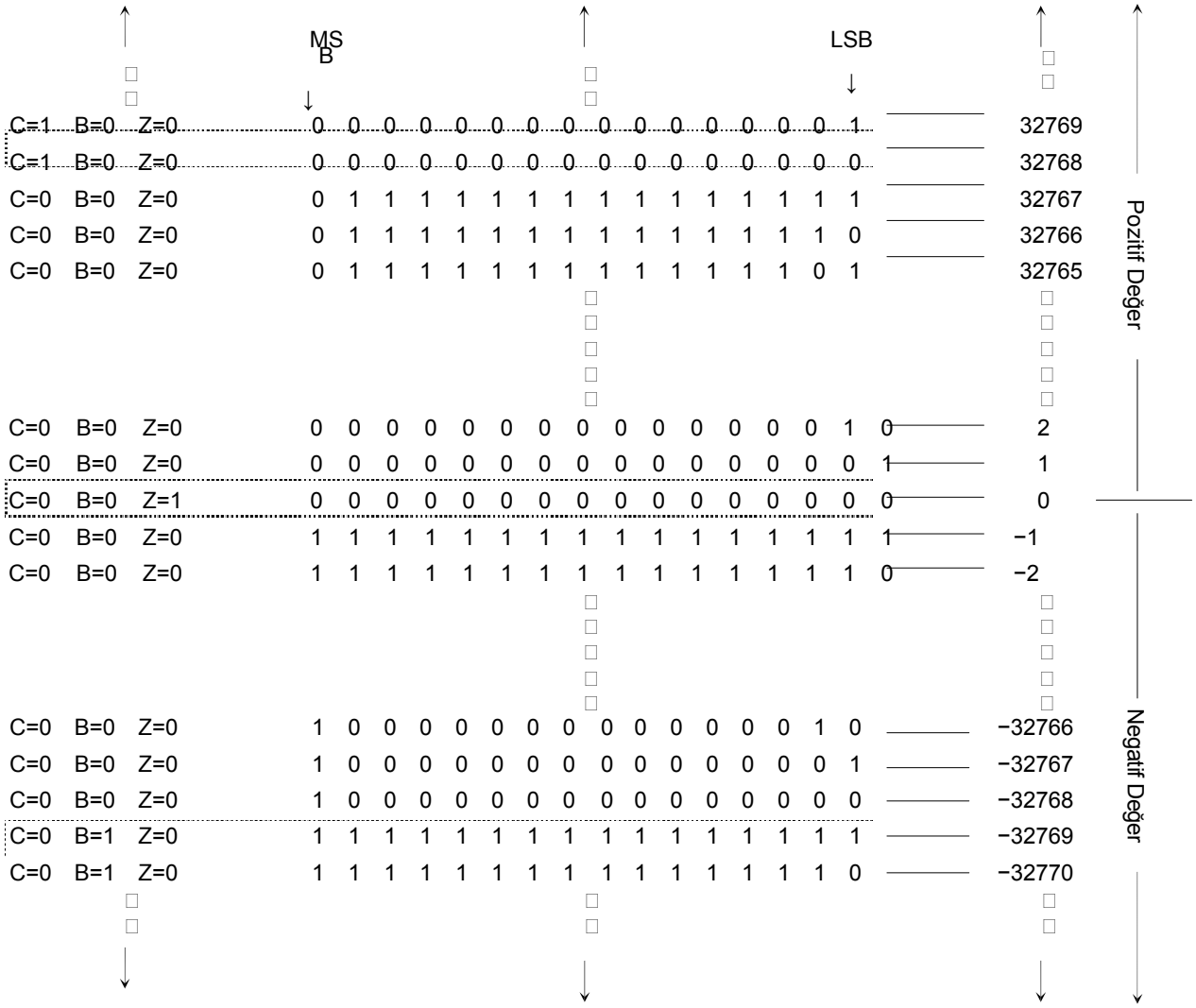
5.5 Toplama/Çıkarmadaki Taşma ve Borç

Pozitif/negatif limiti aşmak PLC de overflow / underflow şeklinde ifade edilir. Sonuçta overflow / underflow'un bir bayrağı tanıtılmıştır. Taşma/Borç bayrağı overflow / underflow' dan farklıdır. Önce, iki operand toplanmalı (çıkarılmalıdır) burada bir toplam (fark) ve birde taşma/borç bayrağı elde edilmelidir. Çünkü numaraların bitlerinin numarası eklenmiş (çıkarılmış), ekli (çıkan) ve toplam (fark) aynıdır, eklemenin (çıkarmanın) sonucunda 16-bit veya 32-bit aşmış toplam değer oluşabilir. Bu yüzden, güncel değeri simgelemekte toplam (fark) operandi ile koordinasyonda olan taşma/borç bayrağı kullanmak gerekmektedir. Toplam (fark) operandinin pozitif limitini aşan sonuçlar eklendiğinde (çıkarıldığında) carry bayrağı aktif olur. Toplam (fark) operandin negatif limitini ((−32768 or −2147483648)) aşan sonuçlar eklendiğinde (çıkarıldığında) borç bayrağı aktif olur. Ancak, eklendikten (çıkarıldıktan) sonra güncel sonuç toplam (fark) operandin elde/borç ile toplanmış değerine eşittir. FBs-PLC ekleme/çıkarma komutunun FO'su güncel sonuç sağlamak için elde ve borç bayraklarının her ikisine de sahiptir.



Tüm Fbs-PLC sayısal operasyonları 2'ye tamamlayanı kullanırken, eklendikten (çıkarıldıktan) sonra elde edilen toplam (fark)'ın negatif değerinin gösterimi olağan negatif sayı gösteriminden farklıdır. İşlem sonucu negatif değerdeyken, toplam (fark) uygulananının MSB'deki görünümü 0 olamaz. Pozitif değer 32768 (2147483648) taşma bayrağını simgeler ve negatif değer −32768 (−2147483648) borç bayrağını simgelemektedir.





C = Elde B = Borç Z = Sıfır