

SAYISAL OPERASYONLAR

SAYISAL OPERASYONLAR

Sayısal kumandaların gerçekleştirilmesi için “byte”, “word” veya “doubleword” operasyonlarına, yani birden fazla “bit”i birlikte işleme emirlerine ihtiyaç vardır. Önemli “word” emirleri; taşıma operasyonu, aritmetik operasyonlar, karşılaştırma operasyonları, kod değişimi, kaydırma-döndürme, sıçrama operasyonları ve “word” lojik kombinasyonlarıdır.

Ayrıca çeşitli lojik işlemlerin sonucuna göre gerçekleştirilecek işlemler, örneğin programın çeşitli kodlara ayrılması için kullanılacak emirler de temel sayısal operasyonlara dahil edilmektedir.

VERİ TİPLERİ

Sayısal operasyonların daha iyi anlaşılabilmesi için S7 1200 PLC programlamada kullanılabilecek veri tiplerinin tam olarak bilinmesi gerekir.

- 1-**Temel data tipleri(Elementary Data Types)
- 2-**Birleştirilmiş data tipleri(Complex Data Types)
- 3-**Parametre Tipleri(Parameter Types)

Data tipi	Tanımlama	Uzunluk	Yazılım aralığı
BOOL	"bit"	1 "bit"	TRUE '1', FALSE '0'
BYTE	İşaretsiz tam sayı Binary sayı Oktal sayı Hexadesimal sayı	8 "bit"	0 → 255 2#0 → 2#11111111 8#0 → 8#377 B#16#0 → B#16#FF
CHAR	ASCII kodu	8 "bit"	ASCII Karakter seti; 'A', "a", "?", "-",...
WORD	İşaretsiz tam sayı Binary sayı Oktal sayı Hexadesimal sayı	16 "bit"	0 → 65535 2#0 → 2#1111111111111111 8#0 → 8#177777 W#16#0 → W#16#FFFF , 16#0 → 16#FFFF
DWORD	İşaretsiz tam sayı Binary sayı Oktal sayı Hexadesimal sayı	32 "bit"	0 → 4294967295 2#0 → 2#1111111111111111.....1111 8#0 → 8#377777777777 DW#16#0000_0000 → DW#16#FFFF_FFFF, 16#0000_0000 → 16#FFFF_FFFF
SINT	İşaretili tam sayı Binary sayı (Pozitif) Oktal sayı (Pozitif) Hexadesimal sayı (+)	8 "bit"	-128 → 127 2#0 → 01111111 8#0 → 8#177 16#0 → 16#7F
USINT	İşaretsiz tam sayı Binary sayı Oktal sayı Hexadesimal sayı	8 "bit"	0 → 255 2#0 → 2#11111111 8#0 → 8#377 16#0 → 16#FF
INT	İşaretili tam sayı Binary sayı (Pozitif) Oktal sayı (Pozitif) Hexadesimal sayı (+)	16 "bit"	- 32768 → 32767 2#0 → 2#0111111111111111 8#0 → 8#77777 16#0 → 16#7FFF
UINT	İşaretsiz tam sayı Binary sayı Oktal sayı Hexadesimal sayı	16 "bit"	0 → 65535 2#0 → 2#1111111111111111 8#0 → 8#177777 16#0 → 16#FFFF

DINT	İşaretli tam sayı Binary sayı (Pozitif) Oktal sayı (Pozitif) Hexadesimal sayı (+)	32 "bit"	- 2147483648 → + 2147483647 2#0 → 2#0111111111111111.....11111111 8#0 → 8#177777777777 16#0 → 16#7FFF FFFF
UDINT	İşaretsiz tam sayı Binary sayı Oktal sayı Hexadesimal sayı	32 "bit"	0 → 4294967295 2#0 → 2#1111111111111111.....11111111 8#0 → 8# 377777777777 16#0000_0000 → 16# FFFF_FFFF
REAL	Virgüllü sayı (IEEE 754) Virgüllü sayı	32 "bit"	-3.402823e+38 → -1.175 495e-38 (Üs'lü gösterim) ±0 +1.175 495e-38 → +3.402823e+38 (Noktalı sayı)
LREAL	Virgüllü sayı (IEEE 754) Virgüllü sayı	64 "bit"	-1.7976931348623158e+308 → -2.2250738585072014e-308 ±0 +2.2250738585072014e-308 → +1.7976931348623158e+308
TIME	IEC Formunda ön işaretli zaman değeri	32 "bit"	T#-24d20h31m23s648ms → T#+24d20h31m23s647ms
DATE	IEC Tarih Hexadesimal sayı	16 "bit"	D#1990-1-1 → D#2168-12-31 W#16#0000 → W16#FF62
TIME_OF_DAY (TOD)	Günün zamanları Hexadesimal sayı	32 "bit"	TOD#00:00:00.000 → TOD#23:59:59.999 DW#16#00000000 → DW#16#05265BFF

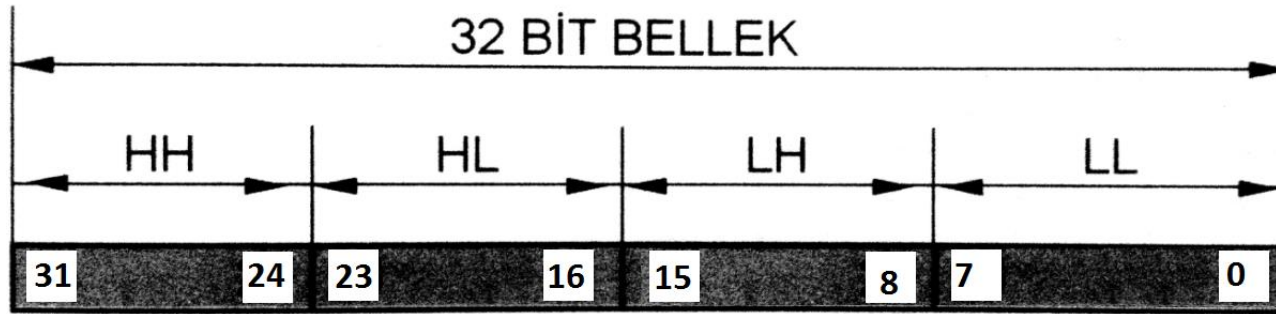
Temel data tipleri (Elementary Data Types)

Data tipleri	Tanımlama	Uzunluk	Yazılım Örneği
DTL	Tarih ve saat (Nano saniye)	96 "bit"	DTL#1970-01-01-00:00:00.0 → DTL#2553-12-31-23:59:59.999999999
STRING	İşaret dizisi	Değişken (254+2)	ASCII kodlu işaretler dizisi 'Yavuz_Eminoglu'
ARRAY	Alan	Değişken	Aynı data tipleri ile oluşturulmuş alanlar
STRUCT	Yapı	Değişken	Değişik data tipleri ile oluşturulmuş alanlar

Birleştirilmiş data tipleri (Complex Data Types)

SIEMENS PLC'LERDE BELLEK YAPILARI

Sayısal operasyonların işlenmesinde genellikle “bit” grupları(“byte”, “word”, Dword”) kullanılır.



32 bit alanın sağ tarafı (0-15.bitler) düşük değerlikli bitlerin yer almasında dolayı düşük “word” (Low Word) olarak tanımlanır. Aynı şekilde sol “word” (16-31.bitler) yüksek “word” (High Word)’dür. Her word’ün kendi içerisinde sağ taraftaki byte’ı düşük byte, sol taraftaki byte’ı yüksek byte’dır. Bu tanımlama sadece PLC’lerde değil bütün işlemci sistemlerinde kullanılır.

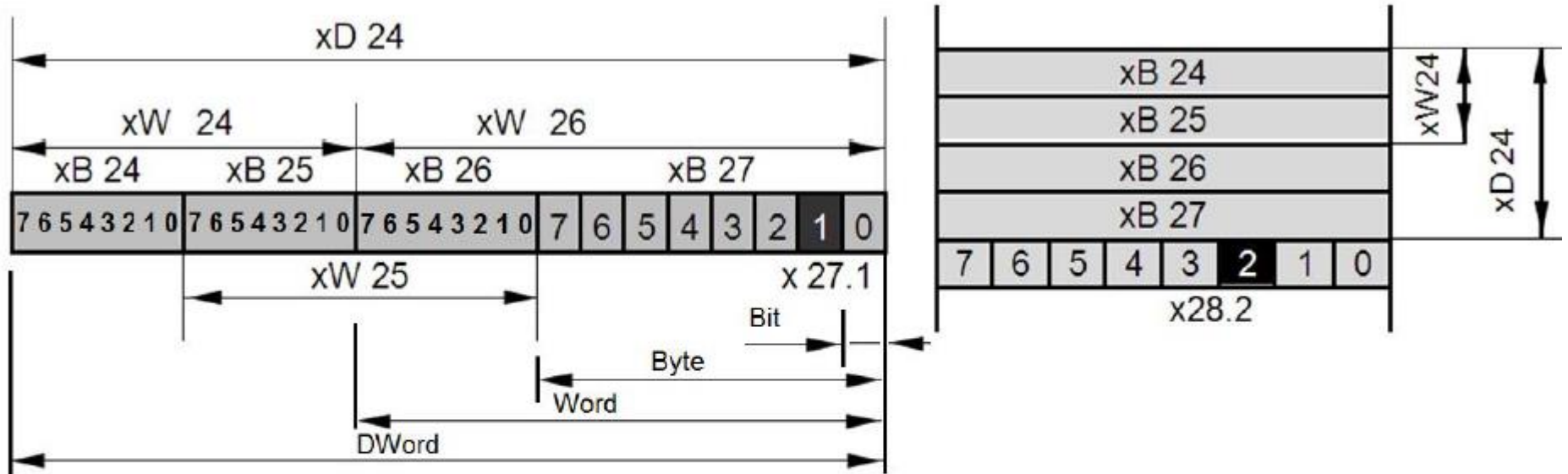
LL(Low Low)=32 bit alanın, düşük word’unun düşük byte’ı

LH(Low High)=32 bit alanın, düşük word’unun yüksek byte’ı

HL(High Low)=32 bit alanın, yüksek word’unun düşük byte’ı

LH(HighHigh)=32 bit alanın, yüksek word’unun yüksek byte’ı

SIEMENS PLC'LERDE ADRES ALANLARININ TANIMLANMASI



x = **I** : Giriş kanalı (Input)
x = **Q** : Çıkış kanalı (Output)
x = **M** : Hafıza elemanı (Merker/Memory)
x = **D** : Veri blok (Data blok)

xW25

x:alan Kodu(I,Q,M,L,D)

W:Alan

uzunluğu("byte","word",Dword")

25:Başlangıç "byte" numarası

SIEMENS PLC'LERDE ADRES ALANLARININ TANIMLANMASI

Siemens PLC'lerde "byte" numaraları soldan sağa, "bit" numaraları sağdan sola doğru artar.(INTEL işlemci bellek yapısı)

Adres alanları kullanılırken ortak alan kullanmamaya özen gösterilmelidir. Örneğin; MW4 ile MW5 alanlarında MB5 ortak alandır.Bir alanın üzerine en son hangi veri yazılmışsa önceki alanın içeriği değişir. Bu nedenle alanlar kullanılırken "BYTE" adresleri 1, WORD adresleri 2, DWORD adresleri 4'er byte'lık uzunluklar olmalı ve ara adresler kullanılmamalıdır. Örneğin MD4 adresi kullanışsa bir sonraki adres MD8 olmalı, MD5,MD6,MD7 kullanılmamalı.

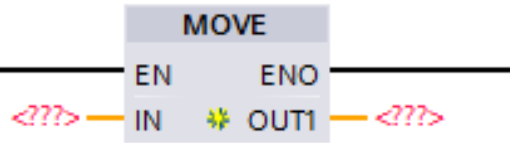
MOVE FONKSİYONU

Bir kaynak adresten okunan (kopyalanan) değerin, istenen hedef adrese yazılmasıdır(yapıştırılmasıdır.)

“Instruction>>Basic Instruction>>Move Operations>>MOVE”

Basic instructions			
Name		Description	Version
▶ +1	Counter operations		
▶ <	Comparator operations		
▶ ±	Math functions		
▼	Move operations		
⌘	MOVE	Move value	
⌘	FieldRead	Read field	
⌘	FieldWrite	Write field	
⌘	MOVE_BLK	Move block	
⌘	UMOVE_BLK	Move block uninterrupt...	
⌘	FILL_BLK	Fill block	
⌘	UFILL_BLK	Fill block uninterruptible	
⌘	SWAP	Swap	

MOVE FONKSİYONU



EN:İşlemin bir şarta bağlanması.Bir lojik kapı bağlantısı yoksa şart yoktur, bağlantı varsa şart vardır ve şart “0” ise işlem gerçekleşmez.

IN:Yükleme hedefi(kopyalanacak alan)

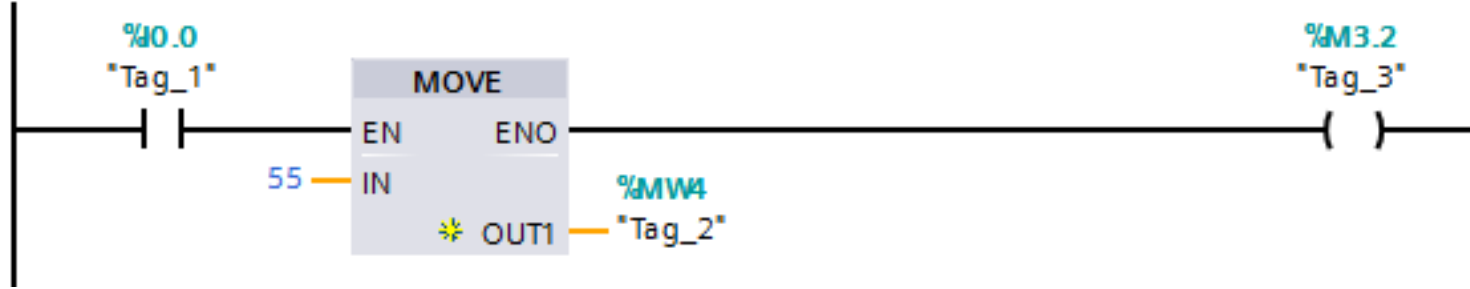
BYTE,WORD,DWORD,SINT,USINT,INT,UINT,DINT,UDINT,TIME,DATE,TOD ,DTL,CHAR,ARRAY,STRUCT,veya STRING içerisindeki karakter.

OUT: Transfer Hedefi(yapıştırılacak alan)

BYTE,WORD,DWORD,SINT,USINT,INT,UINT,DINT,UDINT,TIME,DATE,TOD ,DTL,CHAR,ARRAY,STRUCT,veya STRING içerisindeki karakter.

ENO:İşlemin hatasız yapıldığının “bit” olarak sorgulanması(“EN:0 ise “ “ENO:0”dır. “EN:1” ve işlem hatasız ise “ENO:1” olur.”EN” girişinde değer yok ise “ENO:1” olur.”EN:1” “ENO:0” ise işlem var ama sonuç hatalıdır.

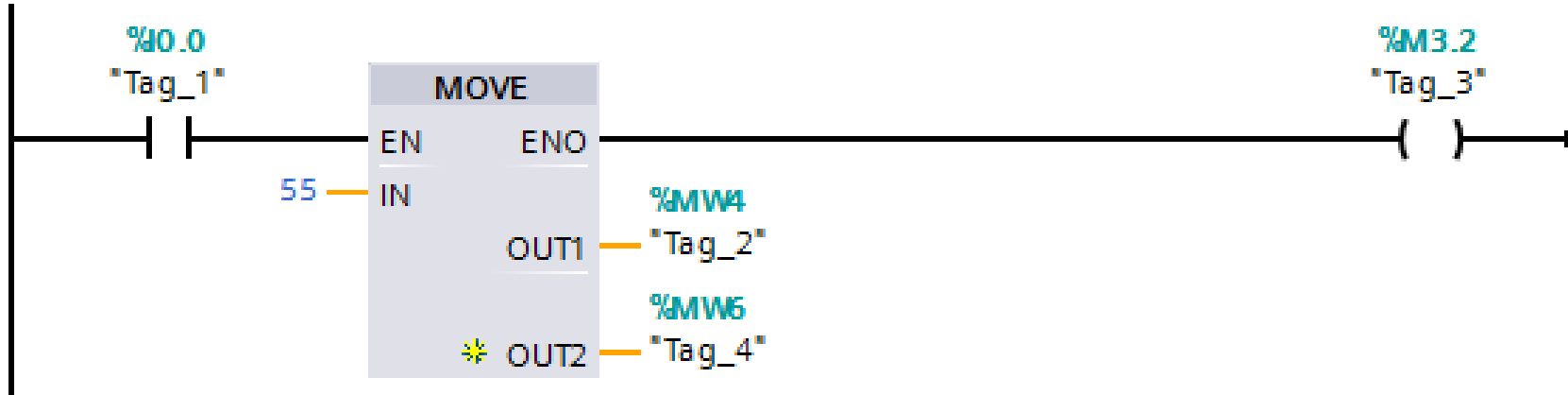
MOVE FONKSİYONU



“EN” girişindeki sinyal **I0.0** uyarıldığında, **IN** girişine yazılan değer (55) veya bellek alanı içeriği **OUT1** çıkışına yazılan bellek alanı (MW4) içerisine yazılır.

Bu arada **“ENO”** çıkış değeri hata olmadığı için uyarılı olur.

MOVE FONKSİYONU



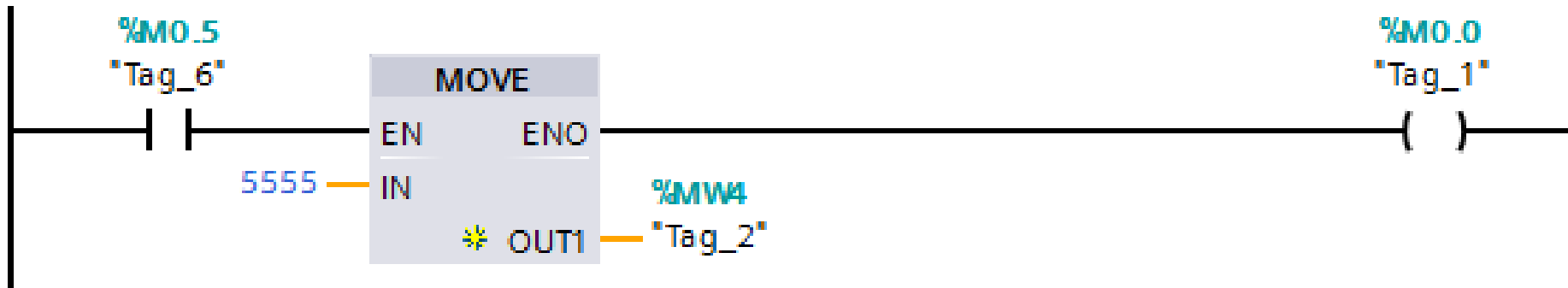
“IN” girişine yazılan veri “OUT” çıkışı çoğaltılarak (*’a basılarak) çok sayıda alana birlikte yazılabilir.

IN girişine yazılan veri uzunluğu OUT çıkışına yazılan veri uzunluğunu aşarsa veri kaybı olur. IN girişine yazılan uzunluk OUT çıkışına yazılan uzunluktan küçükse boş alanlar 0 ile doldurulur.

EN ve ENO fonksiyonları bütün sayısal operasyonlarda aynı özelliklerde kullanılır.

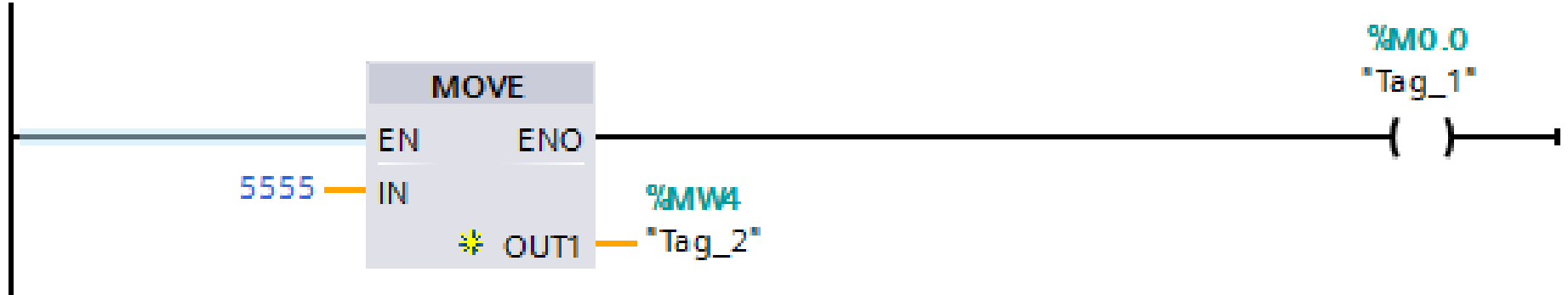
MOVE FONKSİYONU

Alan uzunluğu tanımlanmayan işlemlerde “EN” ve “ENO” durumu.



“EN” şartının gerçekleşmediği her durumda “ENO” çıkışı “0”dır.

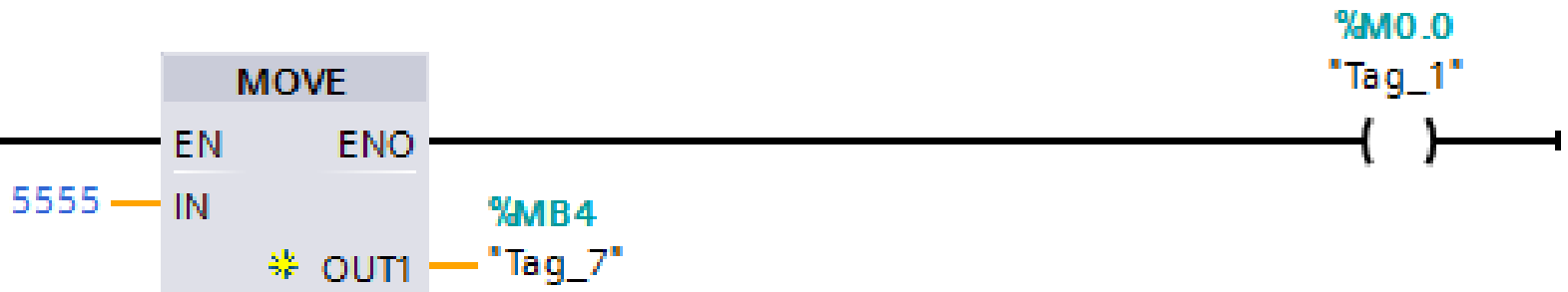
MOVE FONKSİYONU



“EN” şartının gerçekleştiği her durumda, eğer alan uzunluğu tanımlanmamışsa “ENO” çıkışı “1” olur.

5555 sayısı MW4 içerisine atanmış işlemde bir hata olmadığı için ENO çıkışı “1” olur. Bu normal durumdur.

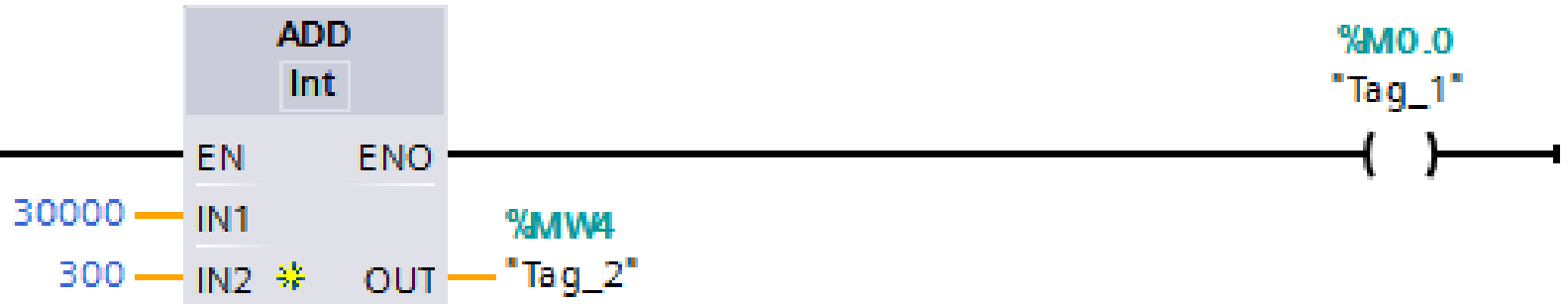
MOVE FONKSİYONU



5555 sayısı MB4'ün içerisine atanmıştır. Normal şartlarda bir WORD uzunluğundaki değer bir "BYTE" içerisine atanmıştır. Yani bir taşma vardır. 16 bit uzunluğundaki sayı 8 bitlik alana sığmaz. Ancak bu işlemde alan uzunluğu (SINT,INT vb.) tanımlanmadığı için, WORD'ün düşük byte'ı içerisindeki değer MB4 içerisine atanmış olur. Bu nedenle hata yok kabul edilir.

MOVE FONKSİYONU

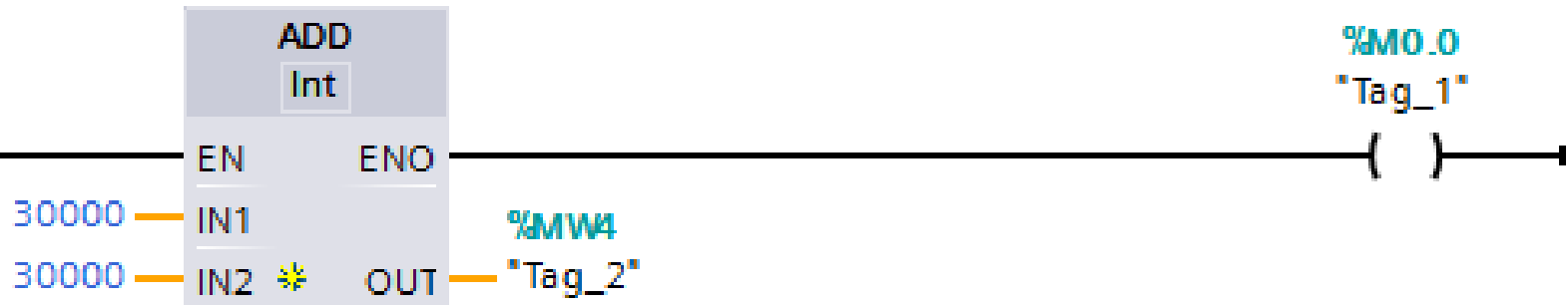
Alan uzunluğu tanımlanan işlemlerde “EN” ve “ENO” durumu.



Alan uzunluğu INT olan bir toplama işlemi yapılmıştır. “30000” ile “300” sayısı toplanmış ve “MW4” içerisine yazılmıştır. Bütün veriler INT uzunluğunda olduğu için sorun yoktur ve “ENO” çıkışı “1”dir.

MOVE FONKSİYONU

Alan uzunluğu tanımlanan işlemlerde “EN” ve “ENO” durumu.

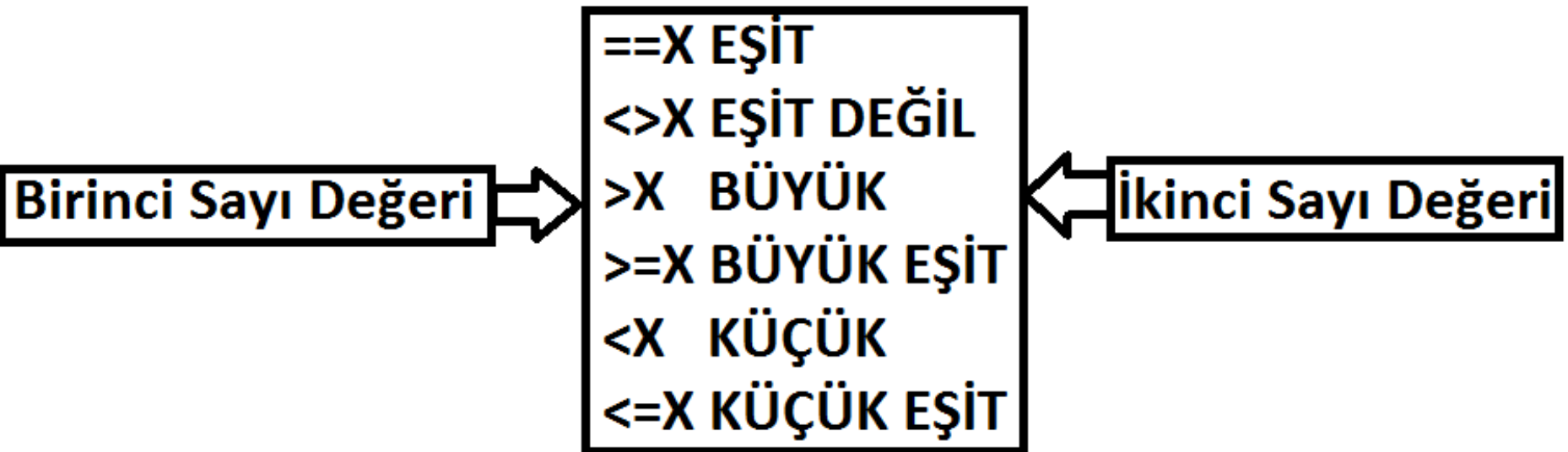


“30000” ile “30000” toplanmış ve MW4 içerisine yazılmıştır. “IN1” ve “IN2” girişine yazılan veriler “INT” uzunluğundadır ve sorun yoktur. Ancak toplama işleminin sonucu “60000” INT’den daha uzundur. Dolayısı ile bir taşma oluşur ve ENO çıkışı “0” olur.











KARŞILAŞTIRMA OPERASYONLARI

STANDART KARŞILAŞTIRICILAR

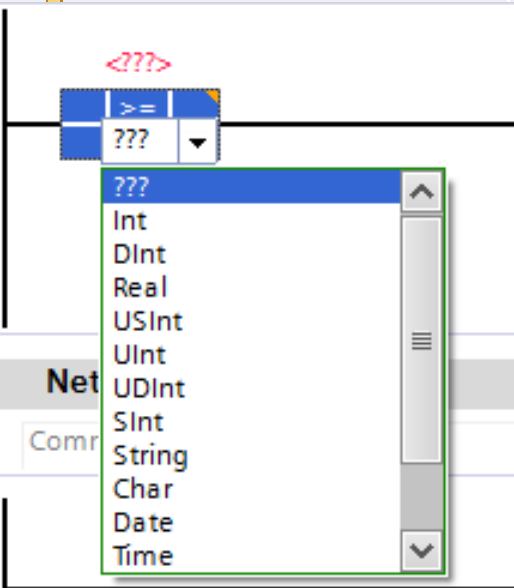
Karşılaştırma operasyonları ile bellek alanlarının içerikleri bir biri ile veya bir bellek alanı içeriğinin bir sayısal değer ile karşılaştırılması yapılır.



KARŞILAŞTIRMA OPERASYONLARI

Comparator operations		
	CMP ==	Equal
	CMP <>	Not equal
	CMP >=	Greater or equal
	CMP <=	Less or equal
	CMP >	Greater than
	CMP <	Less than
	IN_Range	Value within range
	OUT_Range	Value outside range
	- OK -	Check validity
	- NOT_OK -	Check invalidity

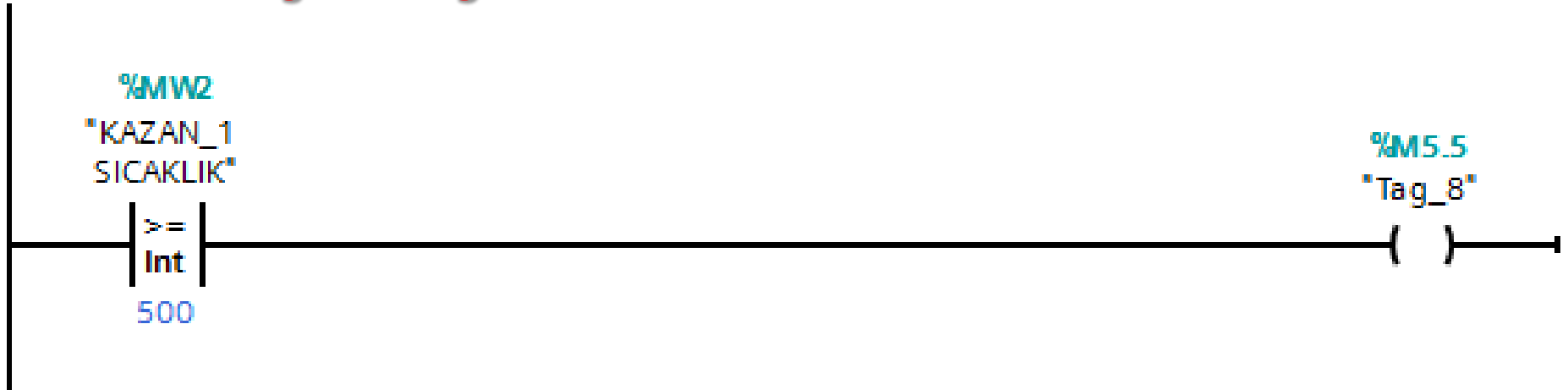
S7 1200 CPU'larda; **aralık içinde "IN_RANGE"**, **aralık dışında "OUT_RANGE"** ve **veri tipi kontrolüne yönelik "OK"** ve **"NOT_OK"** fonksiyonları da kullanılır.



Karşılaştırma tipi ve karşılaştırılacak veri tipi istenilen şekilde değiştirilebilir.

Karşılaştırma fonksiyonları ile işlemciye bir soru sorulmakta ve işlemcide bu soruya "EVET" veya "HAYIR" yanıtı vermektedir. Dolayısı ile karşılaştırma işlemleri sonucunda bilinmesi gereken en önemli nokta ise, operasyon sonucunun bir "bit" ile değerlendirilmesidir.

KARŞILAŞTIRMA OPERASYONLARI



“MW2” (KAZAN_1 SICAKLIK) içeriğinin, “500” tamsayısına eşit veya büyük olup olmadığı sorgulanmaktadır. Eğer MW2 içeriği 500’den büyük veya eşitse bize “EVET” yanıtını verecek yani “M5.5” belleğini “1” yapacaktır. Eğer büyük veya eşit değilse “HAYIR” diyerek “M5.5” belleğini “0” yapacaktır.

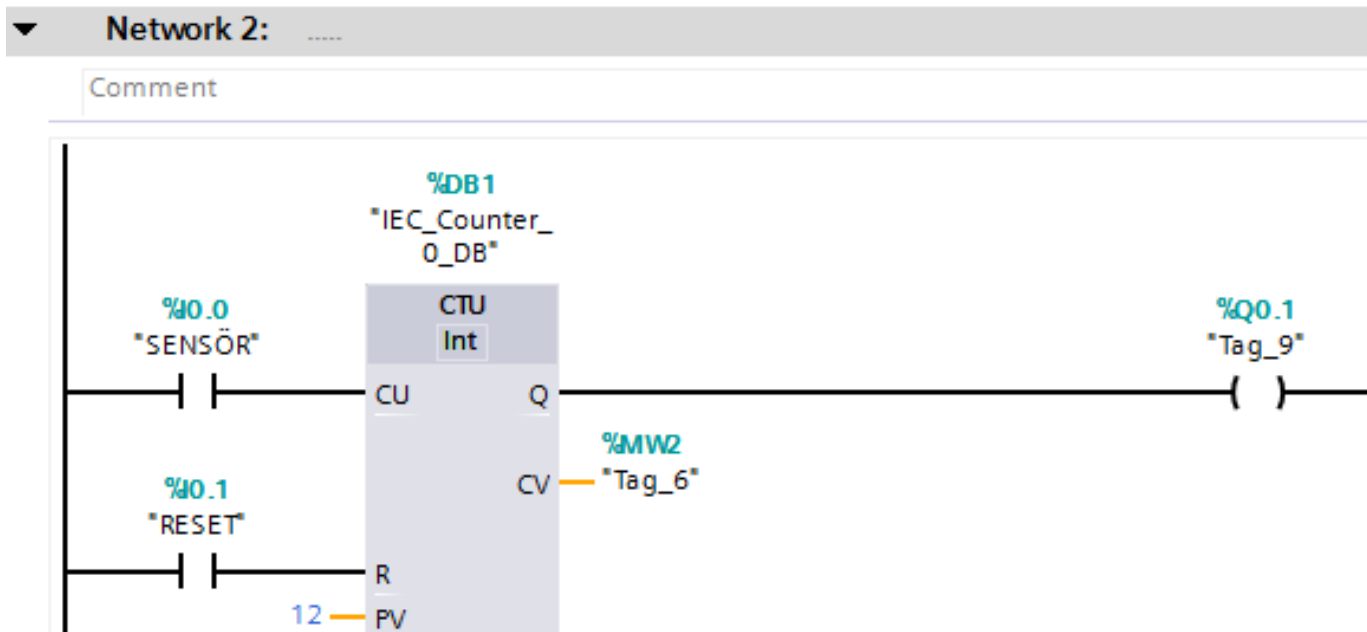
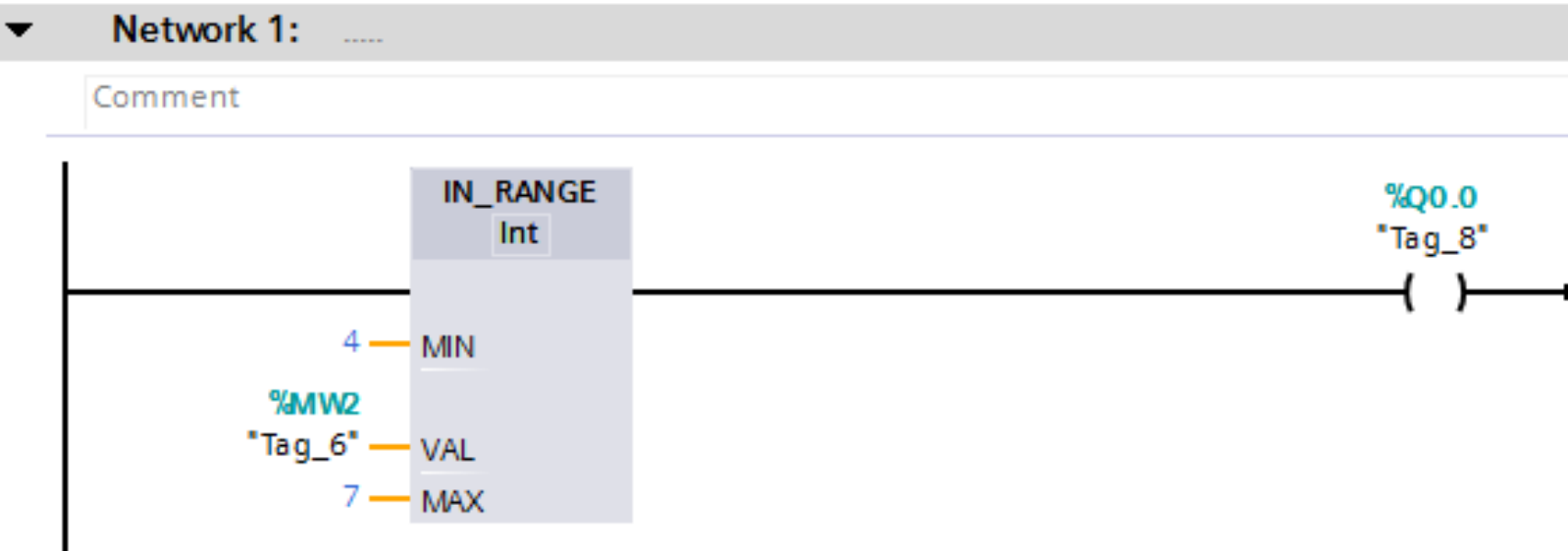
Yukarıdaki mantık bütün karşılaştırma fonksiyonları için aynıdır.

IN_RANGE FONKSİYONU



Sorgulanan veri belirlenen aralıklar içerisinde ise fonksiyonun çıkışı “1” değilse “0”dır. Sorgulanmak istenen veri “VAL” (Value:Değer) girişine yazılır. Tanımlanan aralığa ait alt sınır “MIN”, üst sınır ise “MAX” girişine yazılır. MIN ve MAX parametreleri sayısal değer (Tamsayı,virgüllü sayı) olabileceği gibi bir bellek alanı içeriği de (MW2,MD12,DB1,DBW4) olabilir.

IN_RANGE FONKSİYONU



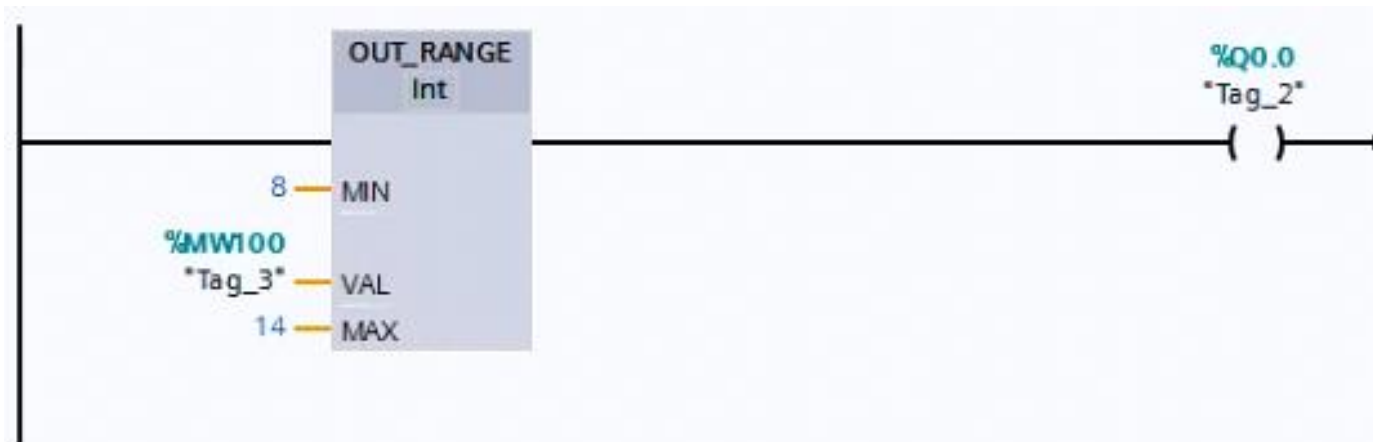
</>

OUT_RANGE FONKSİYONU



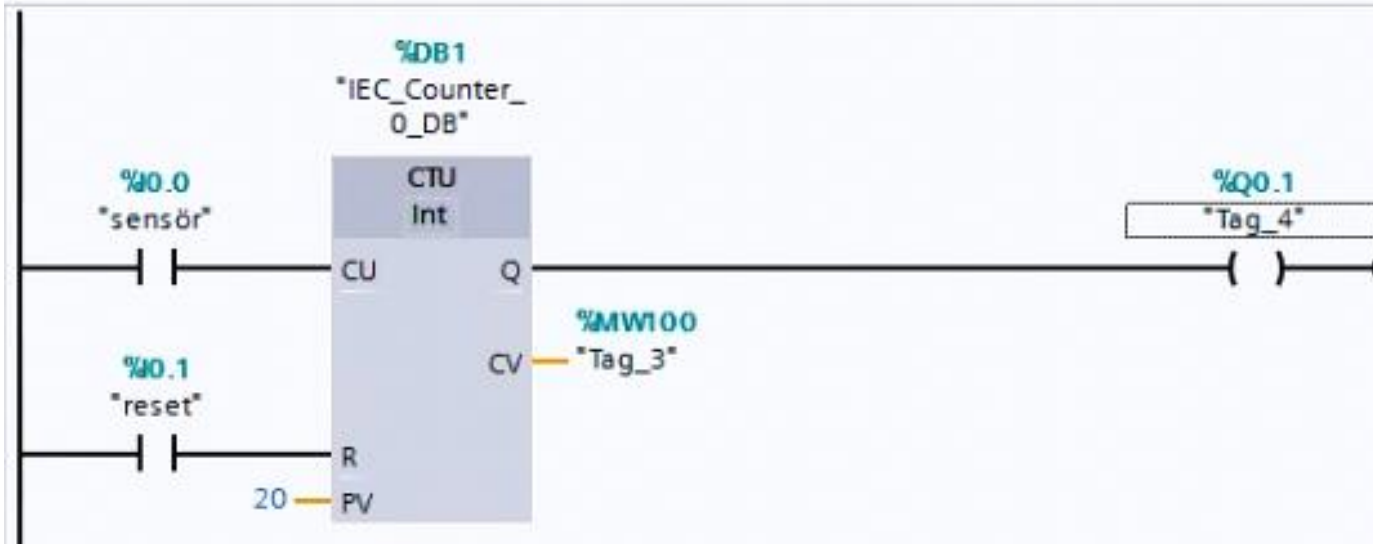
Sorgulanan veri belirlenen aralıkların dışında ise fonksiyonun çıkışı “1” değilse “0”dır. Sorgulanmak istenen veri “VAL” (Value:Değer) girişine yazılır. Tanımlanan aralığa ait alt sınır “MIN”, üst sınır ise “MAX” girişine yazılır. MIN ve MAX parametreleri sayısal değer (Tamsayı,virgüllü sayı) olabileceği gibi bir bellek alanı içeriği de (MW2,MD12,DB1,DBW4) olabilir.

OUT_RANGE FONKSİYONU



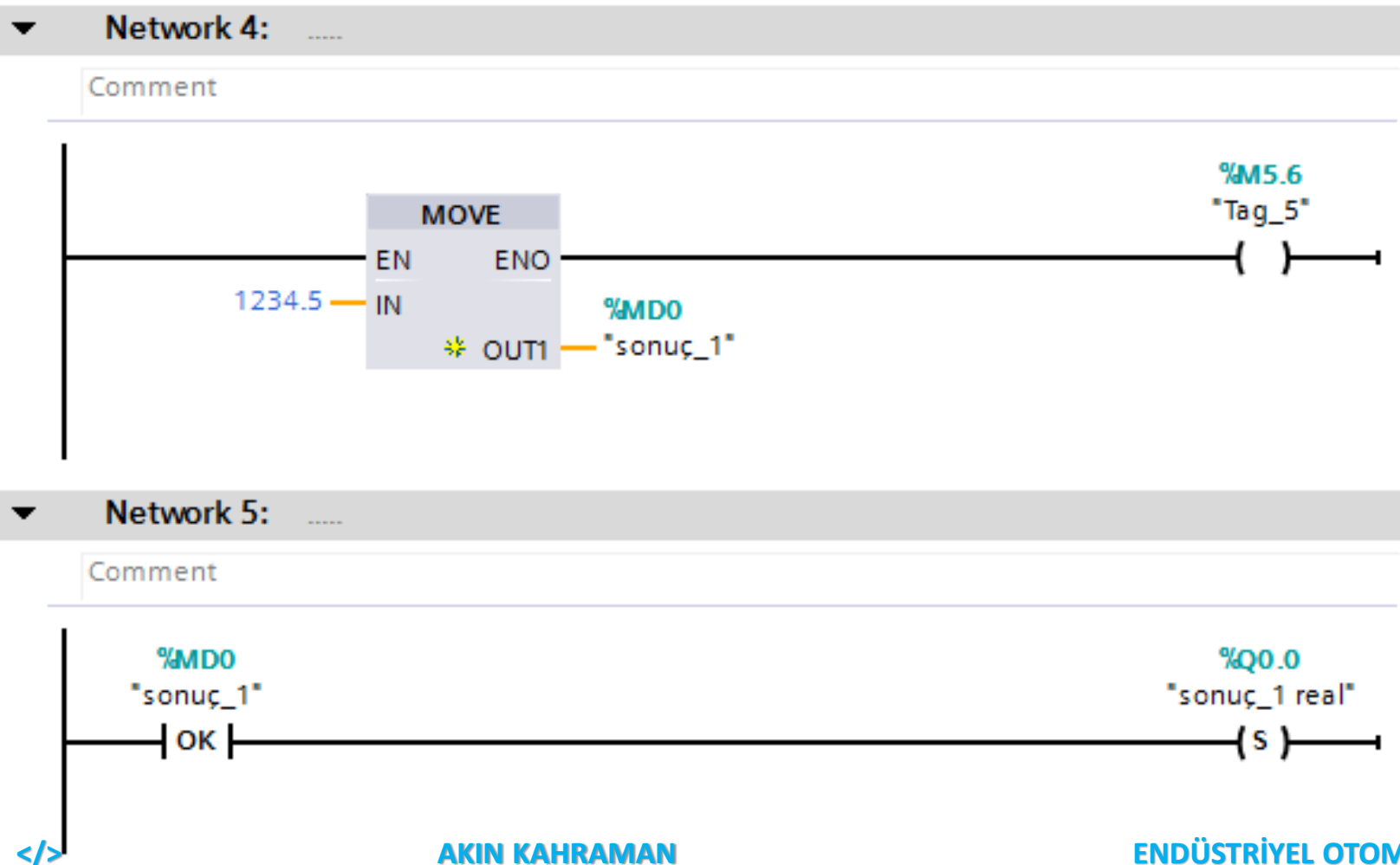
Network 2:

Comment

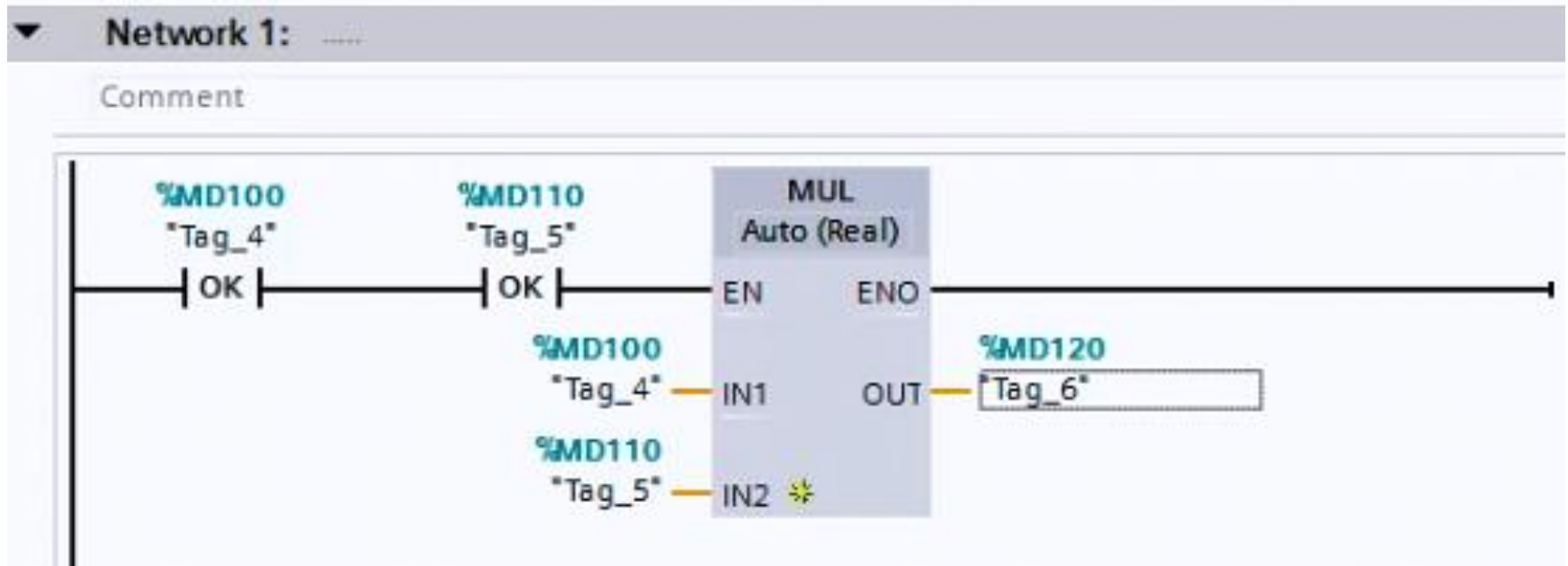


OK FONKSİYONU

Bellek alanı içerisindeki sayının virgüllü olması durumunda çıkış veren fonksiyondur.

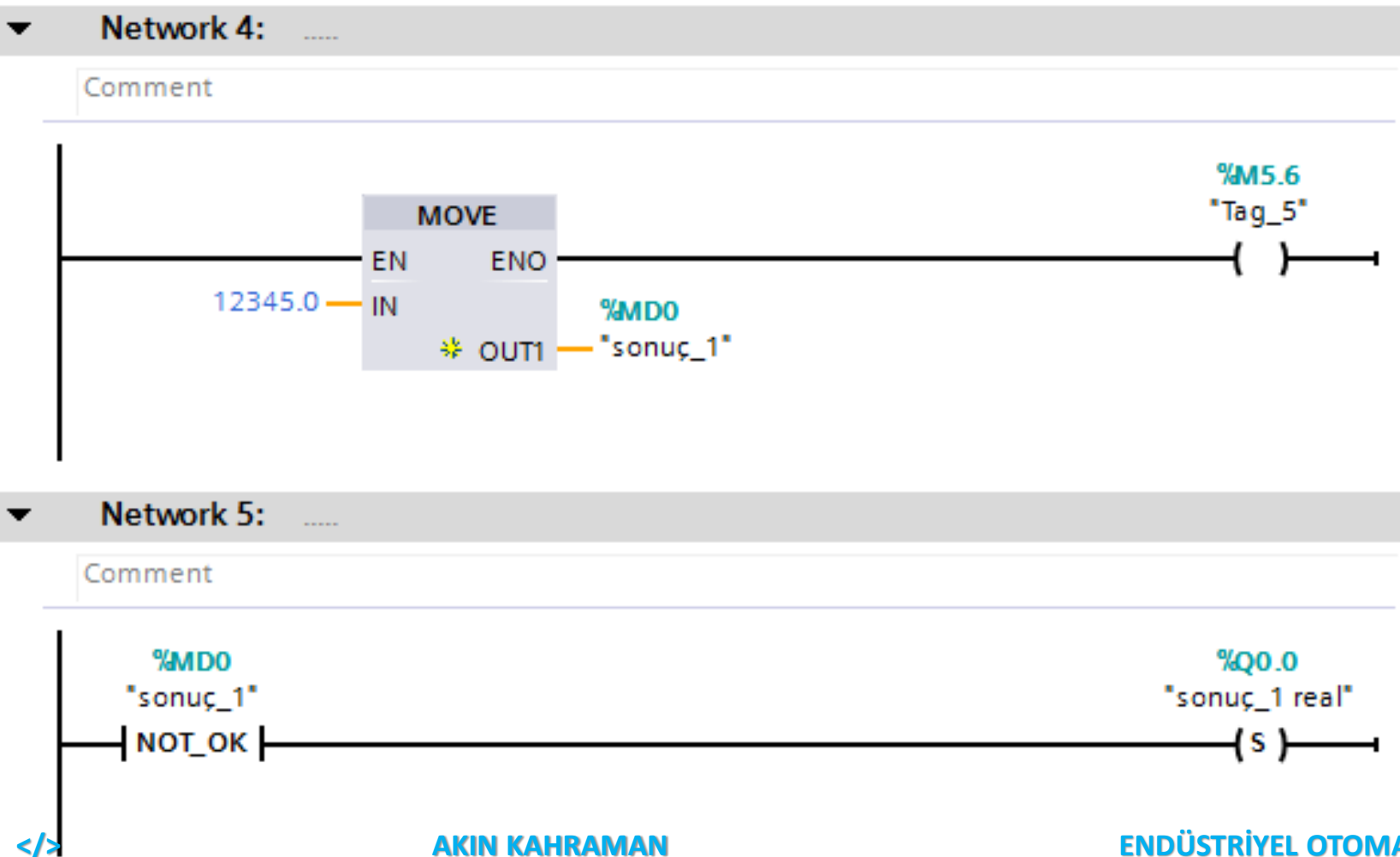


OK FONKSİYONU



NOT_OK FONKSİYONU

Bellek alanı içerisindeki sayının virgüllü olmaması durumunda çıkış veren fonksiyondur.



ÖRNEK: Giriş kanalından uyarılan sinyallerin (OP'den girilen) sayı değeri 10'dan büyük veya eşit, 20'den küçük veya eşit ise çıkış lambası yansın.

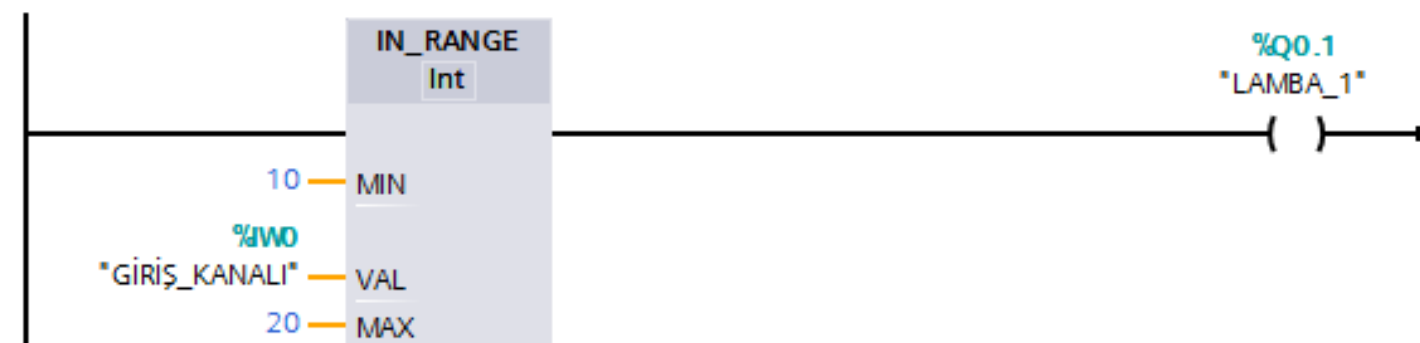
▼ **Network 1: KARŞILAŞTIRMA FONKSİYONLARI İLE**

Comment



▼ **Network 2: IN_RANGE FONKSİYONU İLE**

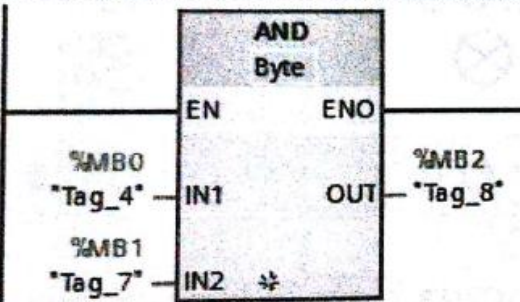
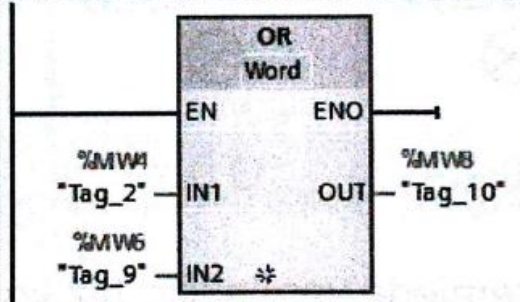
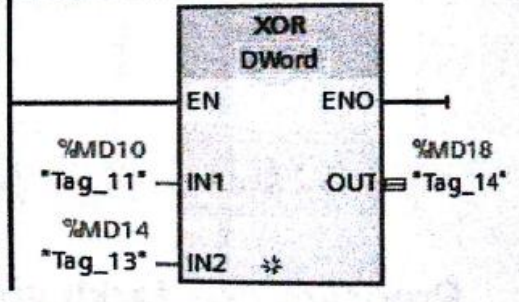
Comment



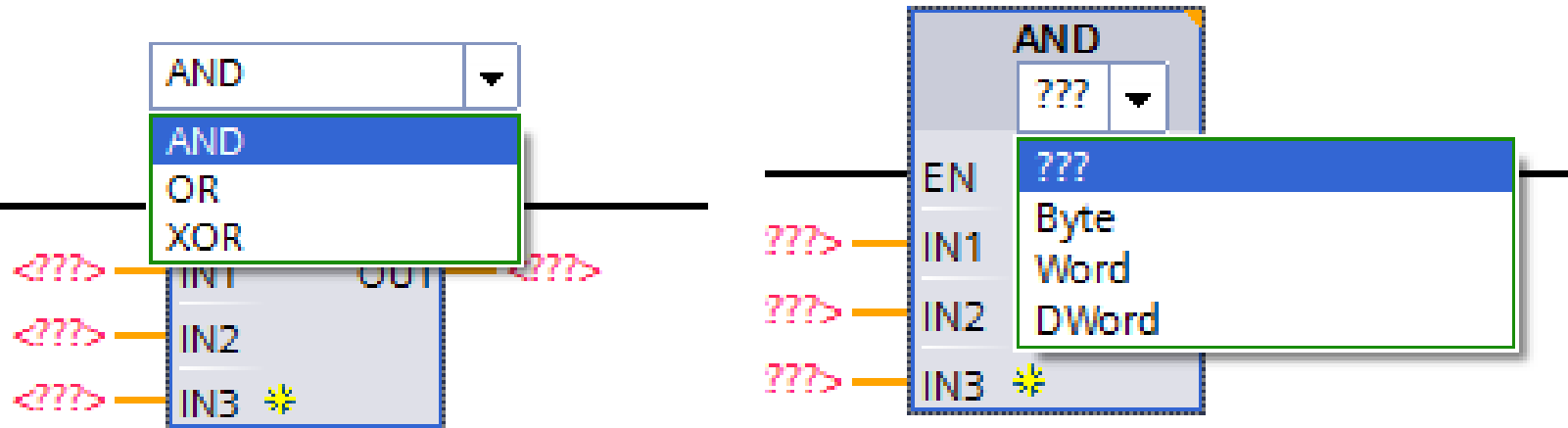
</>

SAYISAL (WORD) LOJİK İŞLEMLERİ

8 “bit”(“Byte”), 16 “bit”(“Word”) veya 32 “bit”(“Doubleword”) ten oluşan bellek alanları içerisindeki aynı numaralı “bit”lerinin **VE,VEYA** ya da **ÖZEL VEYA** işlemlerine tabi tutulmalarıdır.

	“VE” (AND) BYTE, WORD, DWORD	“VEYA” (OR) BYTE, WORD, DWORD	“ÖZEL VEYA” (XOR) BYTE, WORD, DWORD
1.Değer	...1010...	...1010...	...1010...
2.Değer	...1100...	...1100...	...1100...
Sonuç	...1000...	...1110...	...0110...
			

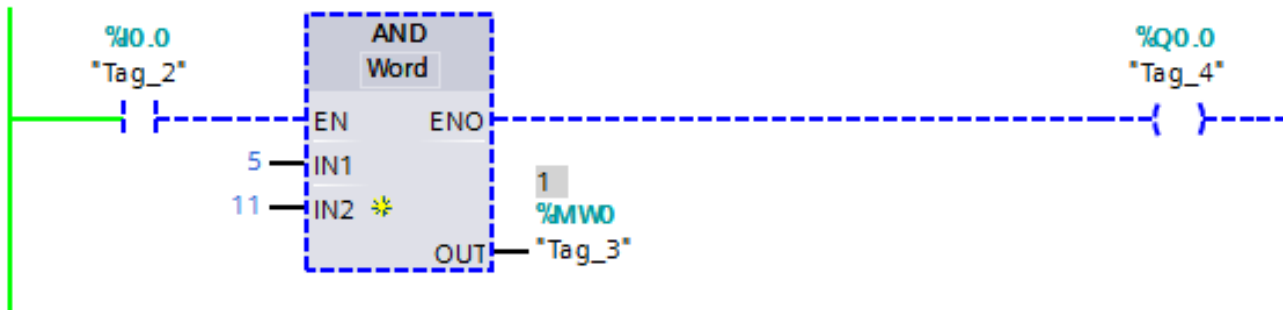
SAYISAL (WORD) LOJİK İŞLEMLERİ



SAYISAL (WORD) LOJİK İŞLEMLERİ

1-MASKELEME

“WORD” lojik operasyon uygulamalarından biri olan maskeleme, istenmeyen bitlerin **“VE”** mantığı ile kapatılmasıdır. **“0”** ile **“VE”** bağlantısı yapılır.

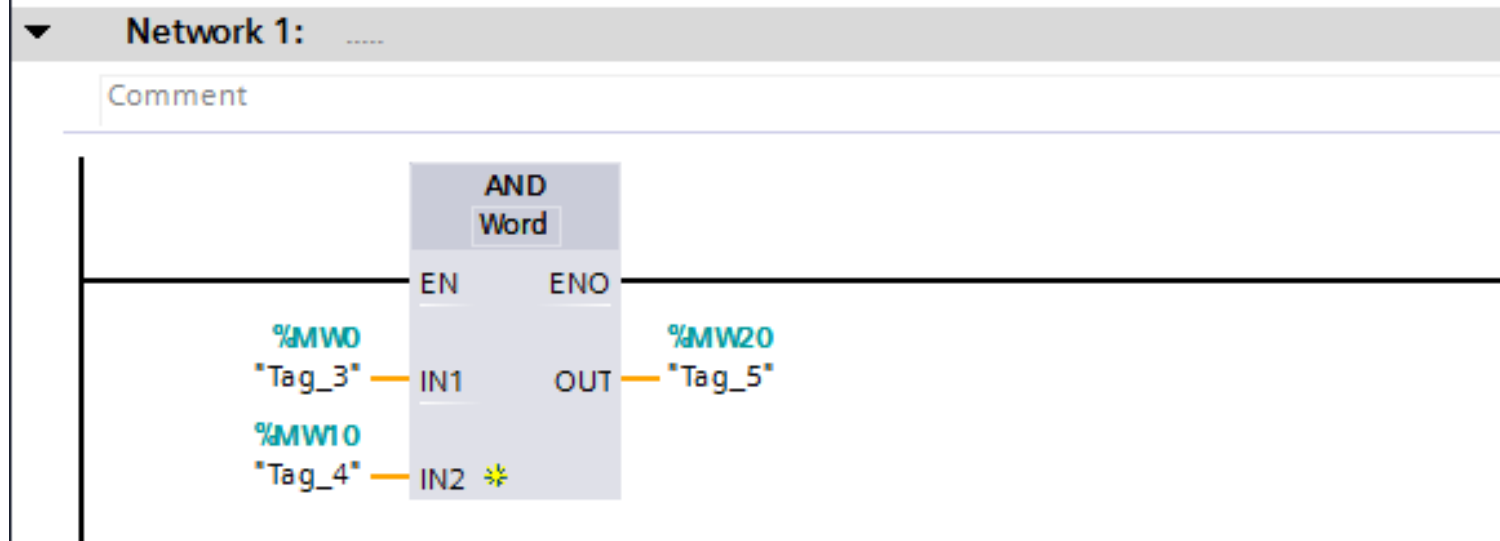


5 =0101

11 =1011

SONUÇ=0001=1

MASKELEME

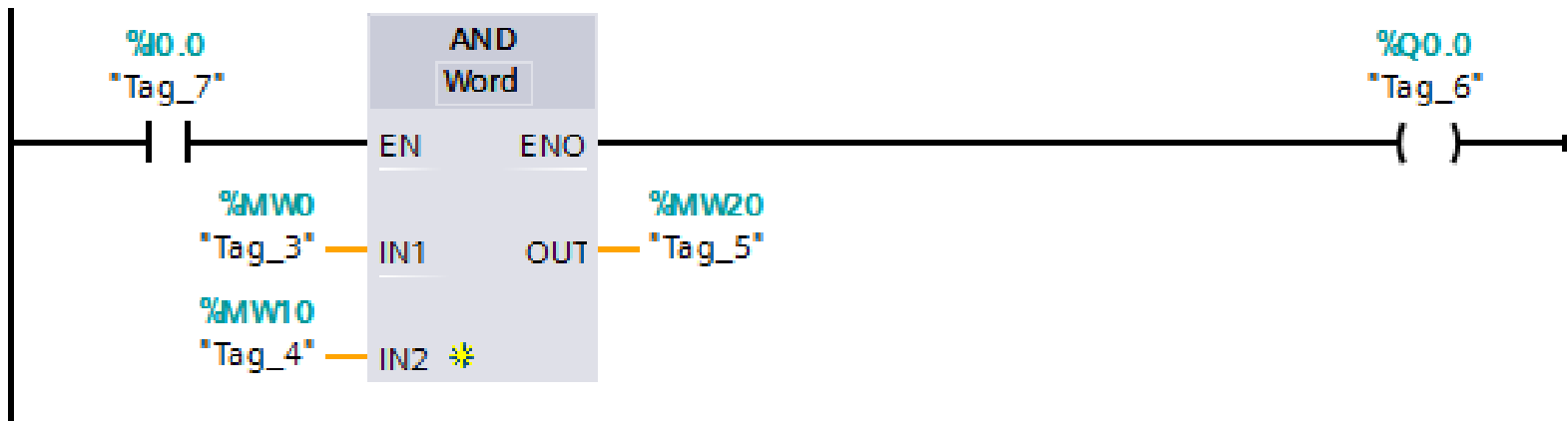


	i	Name	Address	Display format	Monitor value	Force value	F	Comment
1		"Tag_3"	%MW0	Bin	2#0000_0000_0000_0101		<input type="checkbox"/>	
2		"Tag_4"	%MW10	Bin	2#0000_0000_0001_0001		<input type="checkbox"/>	
3		"Tag_5"	%MW20	Bin	2#0000_0000_0000_0001		<input type="checkbox"/>	
4		<Add new>					<input type="checkbox"/>	

MASKELEME

▼ Network 1:

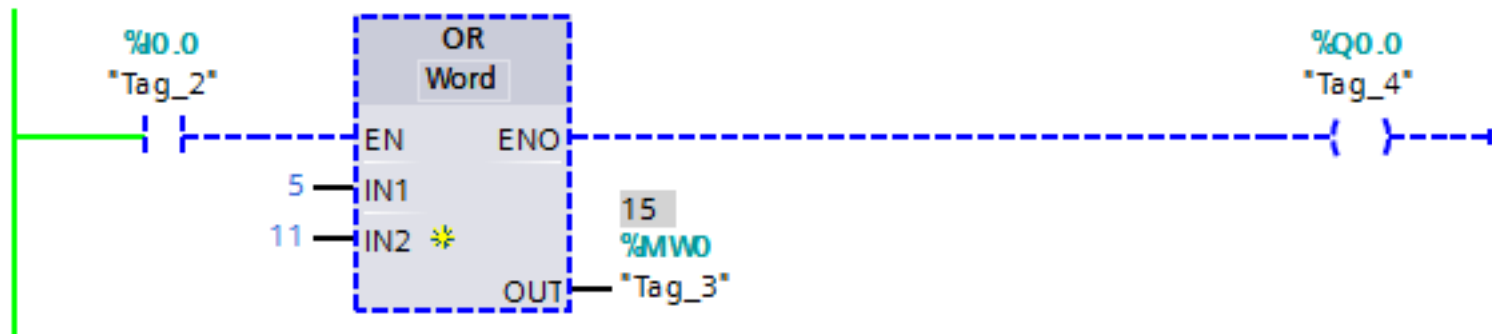
Comment



SAYISAL (WORD) LOJİK İŞLEMLERİ

2-ÜZERİNE YAZMA

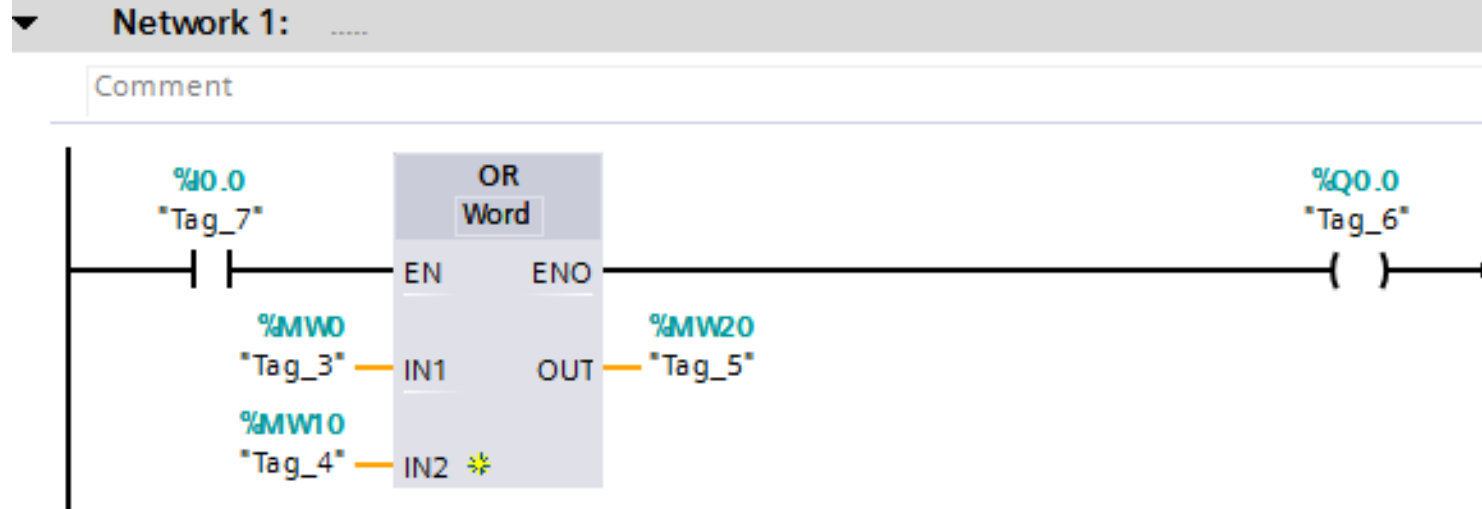
“WORD” lojik operasyon uygulamalarından biri üzerine yazma “VEYA” fonksiyonu ile “1” yazmaktadır. “1” ile “VEYA” bağlantısı yapılır.



5 =0101
11 =1011

SONUÇ=1111=15

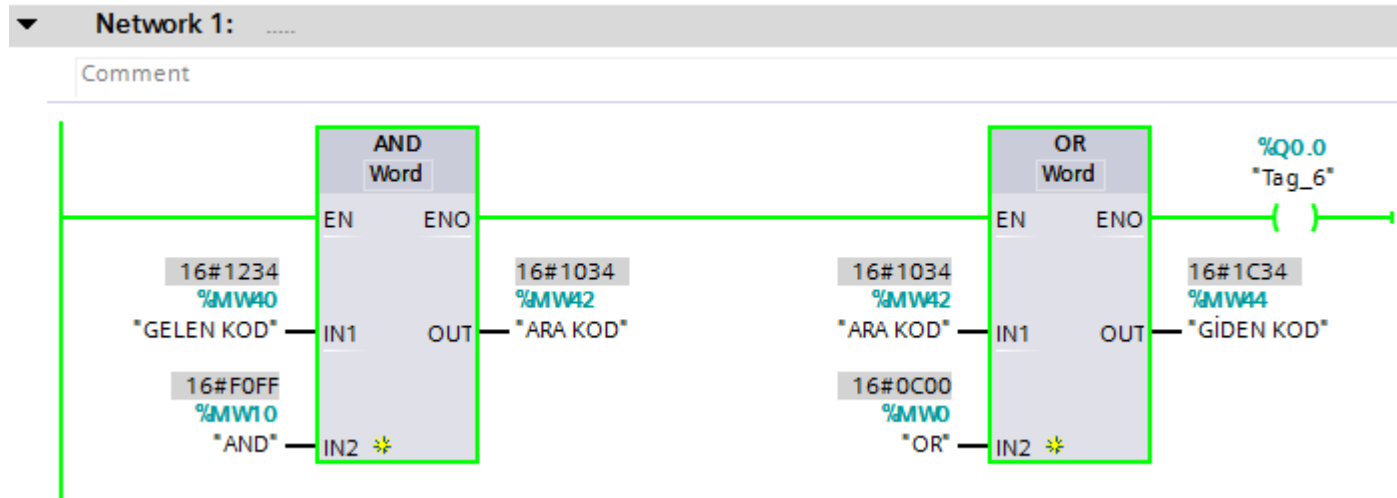
ÜZERİNE YAZMA



	i	Name	Address	Display format	Monitor value	Force value	F	Comment
1		*Tag_3	%MW0	Bin	2#0000_0000_0000_1010		<input type="checkbox"/>	
2		*Tag_4	%MW10	Bin	2#0000_0000_0001_0101		<input type="checkbox"/>	
3		*Tag_5	%MW20	Bin	2#0000_0000_0001_1111		<input type="checkbox"/>	
4		<Add new>					<input type="checkbox"/>	

ÖRNEK

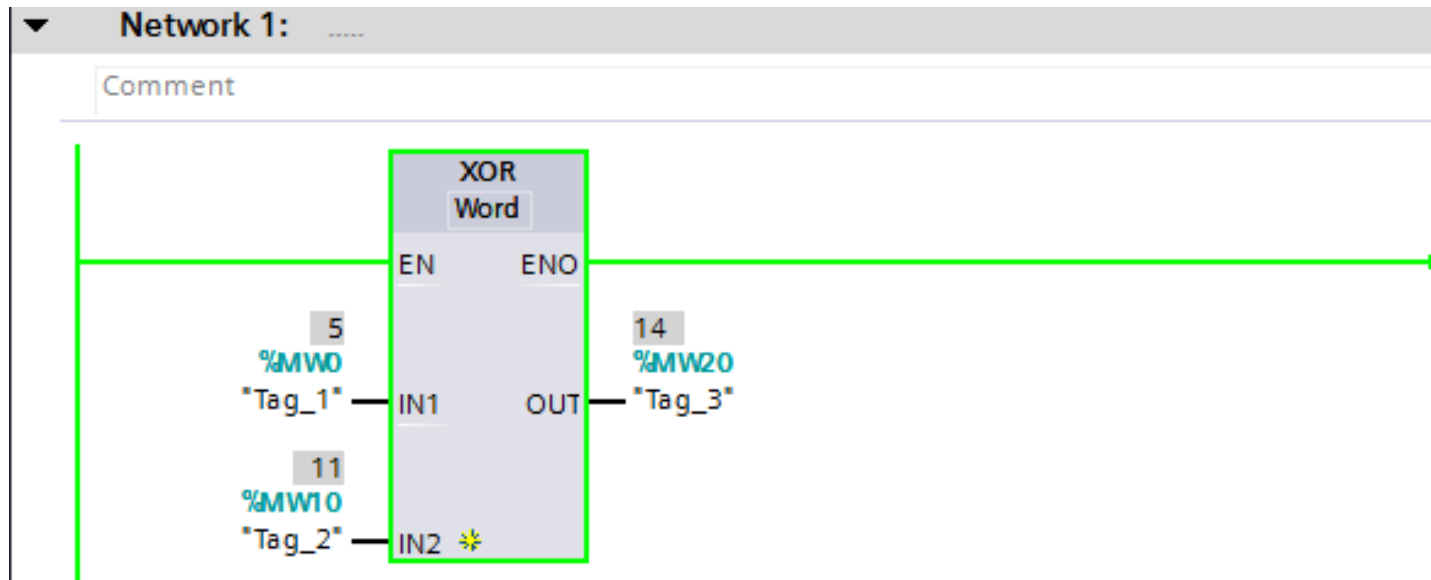
Üretim hattındaki bir ürünün 16 bit uzunluğundaki kodu okunup diğer alanlarına dokunmadan 9-12 bitlerine hex olarak C kodu eklenecektir.



	i	Name	Address	Display format	Monitor value	Fc
1		"GELEN KOD"	%MW40	Hex	16#1234	
2		"GİDEN KOD"	%MW44	Hex	16#1C34	
3		"AND"	%MW10	Hex	16#F0FF	
4		"OR"	%MW0	Hex	16#0C00	
5		"ARA KOD"	%MW42	Hex	16#1034	
6			<Add new>			

SAYISAL (WORD) LOJİK İŞLEMLERİ

3-XOR İŞLEMİ

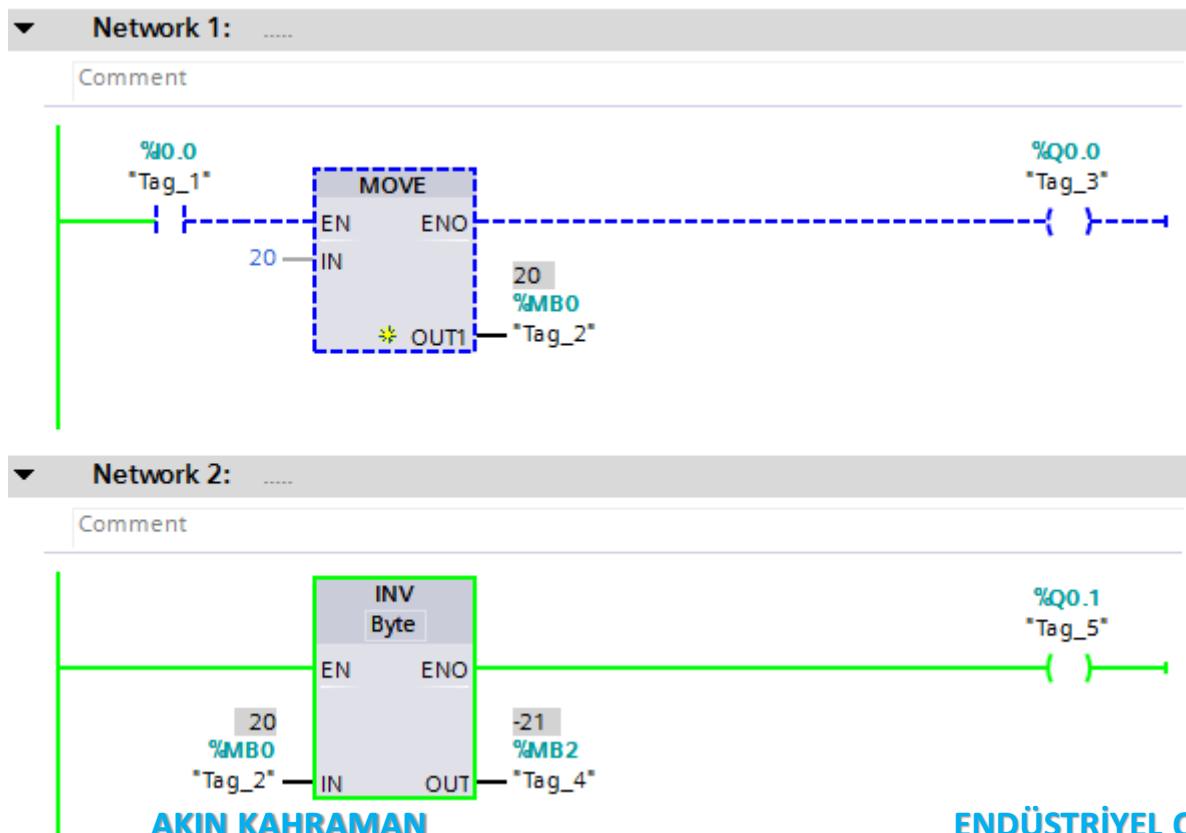


	i	Name	Address	Display format	Monitor value	Force value	F	C
1		"Tag_1"	%MW0	Bin	2#0000_0000_0000_0101		<input type="checkbox"/>	
2		"Tag_2"	%MW10	Bin	2#0000_0000_0000_1011		<input type="checkbox"/>	
3		"Tag_3"	%MW20	Bin	2#0000_0000_0000_1110		<input type="checkbox"/>	

SAYISAL (WORD) LOJİK İŞLEMLERİ

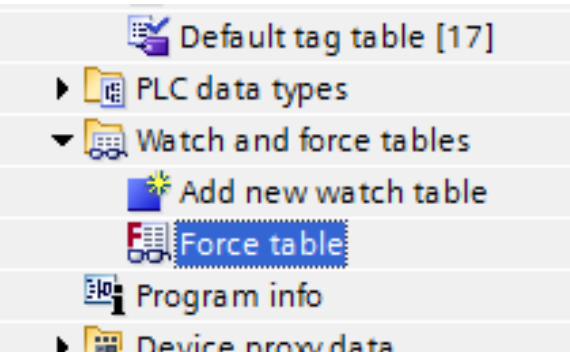
3-KOMPLEMENT ALMA (INV)

Bir bellek alanı içerisindeki her bir bit 'in tek tek **tersinin** alınmasıdır. Veriyi meydana getiren bütün "0"ların "1", "1"lerin "0" yapılmasıdır.



SAYISAL (WORD) LOJİK İŞLEMLERİ

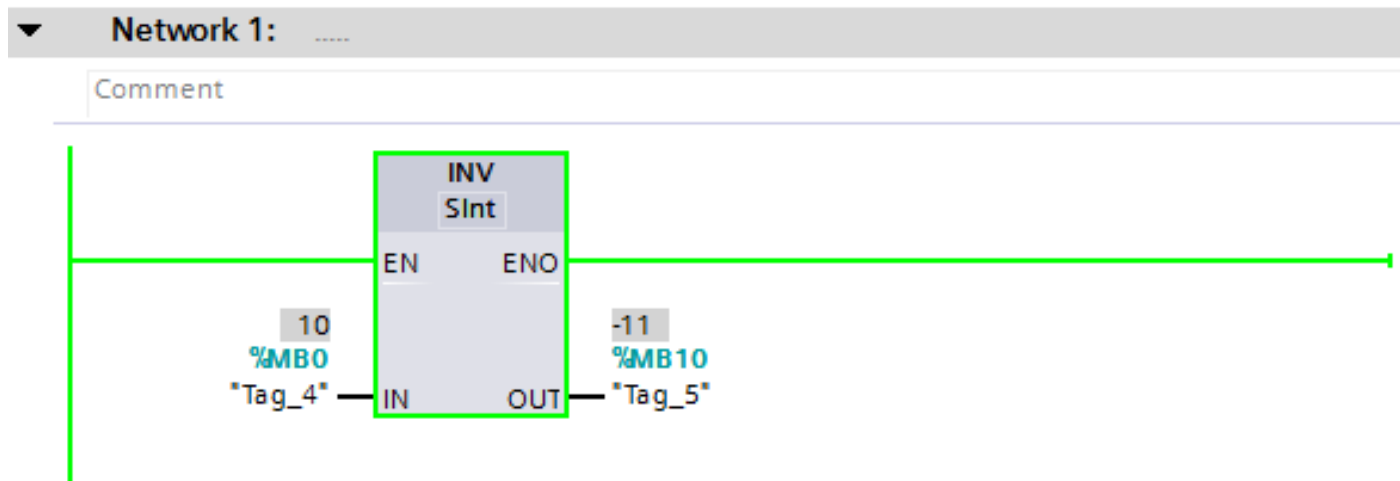
3-KOMPLEMENT ALMA (INV)



	i	Name	Address	Display format	Monitor value	Force value	F	Comment
1		"Tag_2"	%MB0	Bin	2#0001_0100		<input type="checkbox"/>	
2		"Tag_4"	%MB2	Bin	2#1110_1011		<input type="checkbox"/>	
3		<Add new>					<input type="checkbox"/>	

Yapılan işlemler **“Watch and force tables”** altındaki **“Force table”** içinden izlenebilir.

3-KOMPLEMENT ALMA (INV)

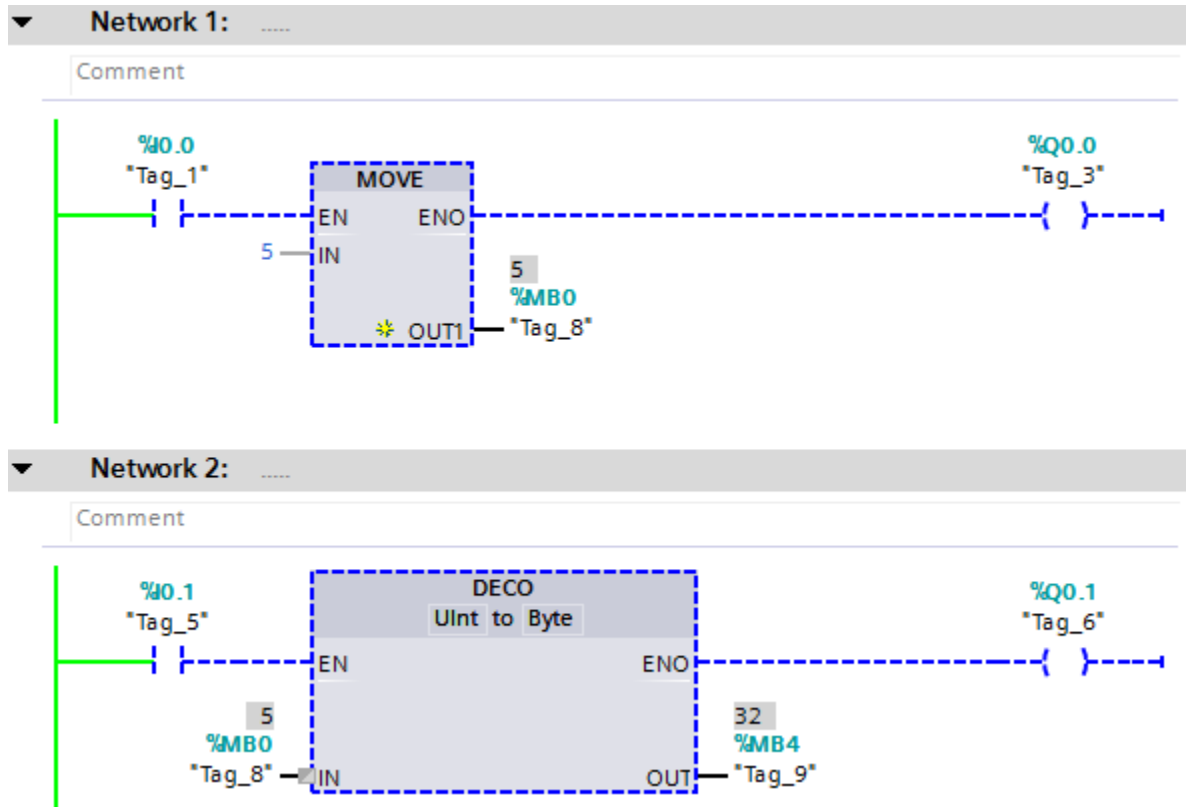


	i	Name	Address	Display format	Monitor value	For
1		*Tag_4*	%MB0	Bin	2#0000_1010	
2		*Tag_5*	%MB10	Bin	2#1111_0101	
3			<Add new>			

SAYISAL (WORD) LOJİK İŞLEMLERİ

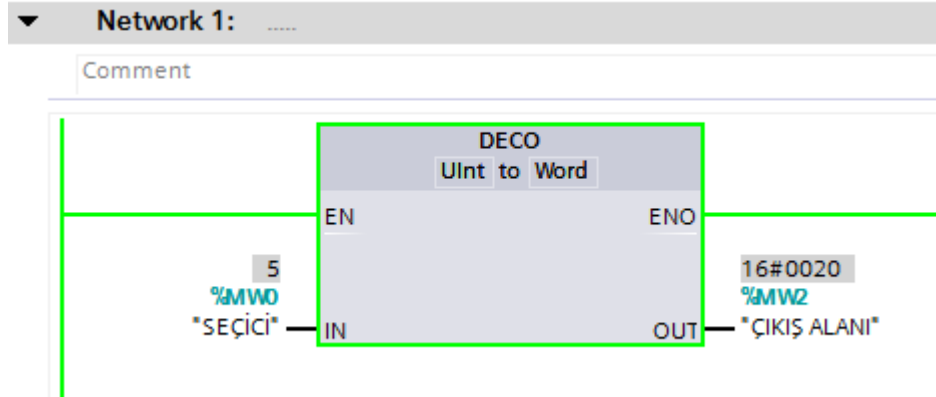
4-KOD ÇÖZÜCÜ (DECO)

"IN" girişine uygulanan bellek alanının desimal sayı değeri, **"OUT"** çıkışına yazılan bellek alanının o numaralı bitini **"1"** yapar.



	i	Name	Address	Display format	Monitor value	Force value	F	Comment
1		"Tag_8"	%MB0	Bin	2#0000_0101		<input type="checkbox"/>	
2		"Tag_9"	%MB4	Bin	2#0010_0000		<input type="checkbox"/>	
3		<Add new>					<input type="checkbox"/>	
AKIN KAHRAMAN					ENDÜSTRİYEL OTOMASYON			

4-KOD ÇÖZÜCÜ (DECO)



	i	Name	Address	Display format	Monitor value
1		"SEÇİCİ"	%MW0	DEC	5
2		"ÇIKIŞ ALANI"	%MW2	Bin	2#0000_0000_0010_0000
3			<Add new>		

SAYISAL (WORD) LOJİK İŞLEMLERİ

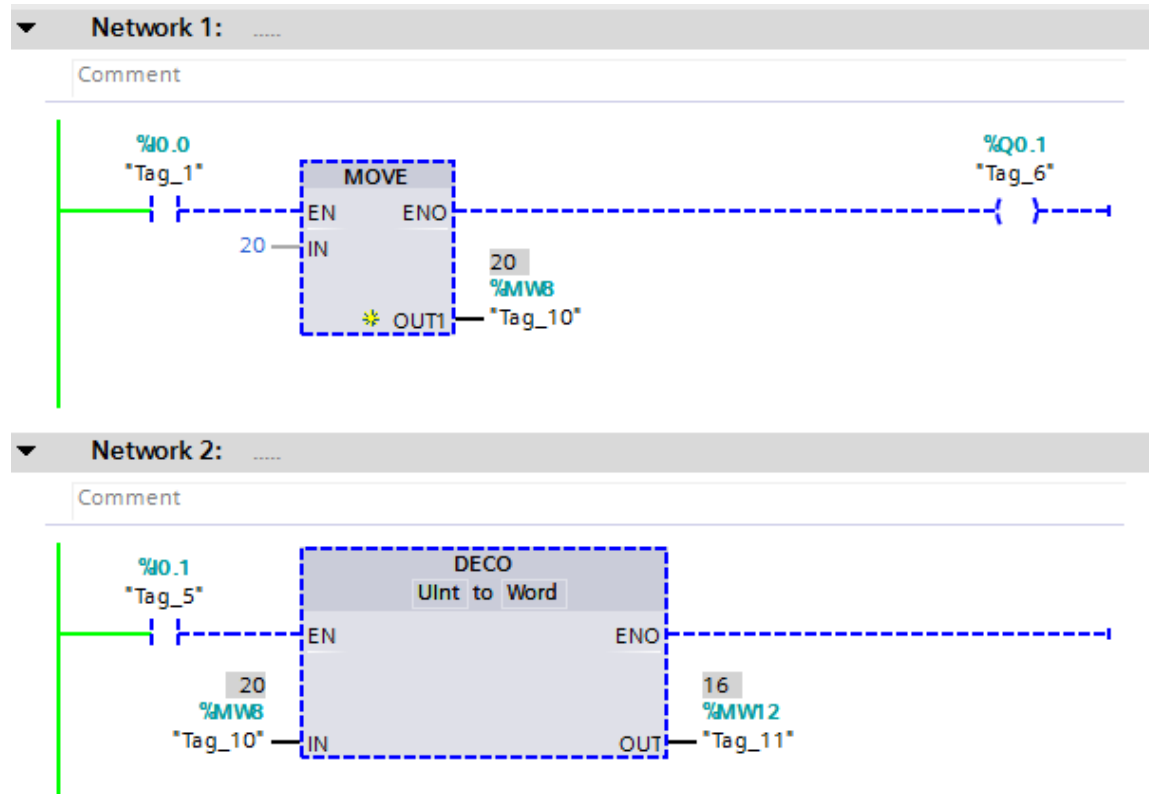
4-KOD ÇÖZÜCÜ (DECO)

“IN” girişine, çıkışa yazılan bellek alanı “bit” sayısından dahi büyük sayı girilirse, modülo bölme işlemi uygulanır. Çıkıştaki bellek alanı “bit” sayısı, giriş değerine bölünür. Bölme sonrasında kalan sayı numaralı “bit” “1 ” olur.

SAYISAL (WORD) LOJİK İŞLEMLERİ

4-KOD ÇÖZÜCÜ (DECO)

Aşağıdaki uygulamada 20 sayısı bir “word”’lük bit sayısına (16) bölünmüş ve kalan 4 sayısı çıkışa yazılmıştır.

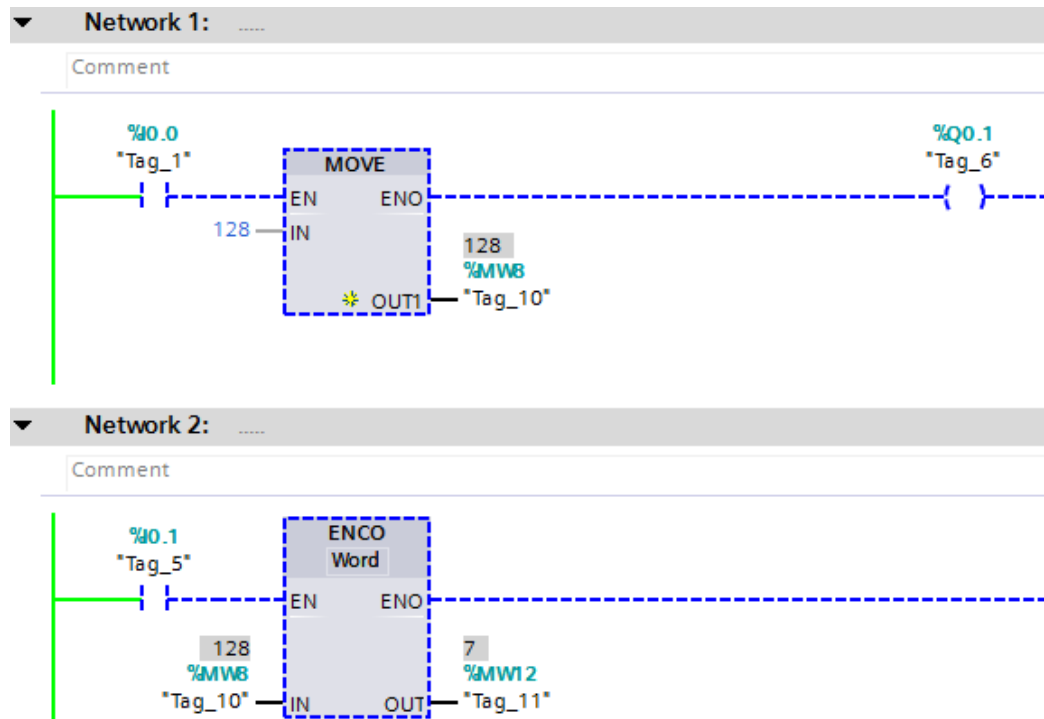


	i	Name	Address	Display format	Monitor value	Force value	F	Comment
1		*Tag_10*	%MW8	DEC+/-	20		<input type="checkbox"/>	
2		*Tag_11*	%MW12	Bin	2#0000_0000_0001_0000		<input type="checkbox"/>	

SAYISAL (WORD) LOJİK İŞLEMLERİ

5-KODLAYICI (ENCO)

"IN" girişindeki bellek alanı veya sayıya ait uyarılı olan bitlerden en küçük numaralı bit'in numarasını **OUT** çıkışına yazar.



	i	Name	Address	Display format	Monitor value	Force value	F	Comment
1		"Tag_10"	%MW8	Bin	2#0000_0000_1000_0000		<input type="checkbox"/>	
2		"Tag_11"	%MW12	Bin	2#0000_0000_0000_0111		<input type="checkbox"/>	

SAYISAL (WORD) LOJİK İŞLEMLERİ

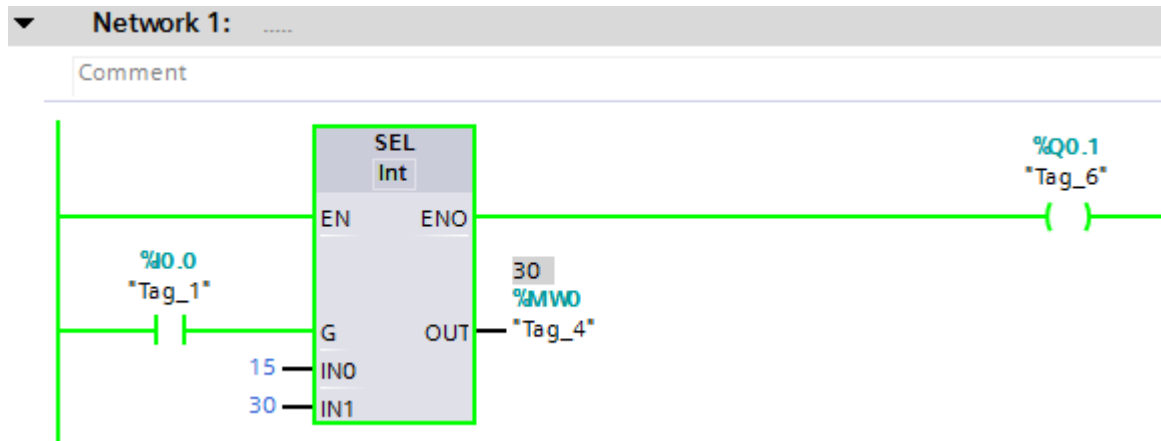
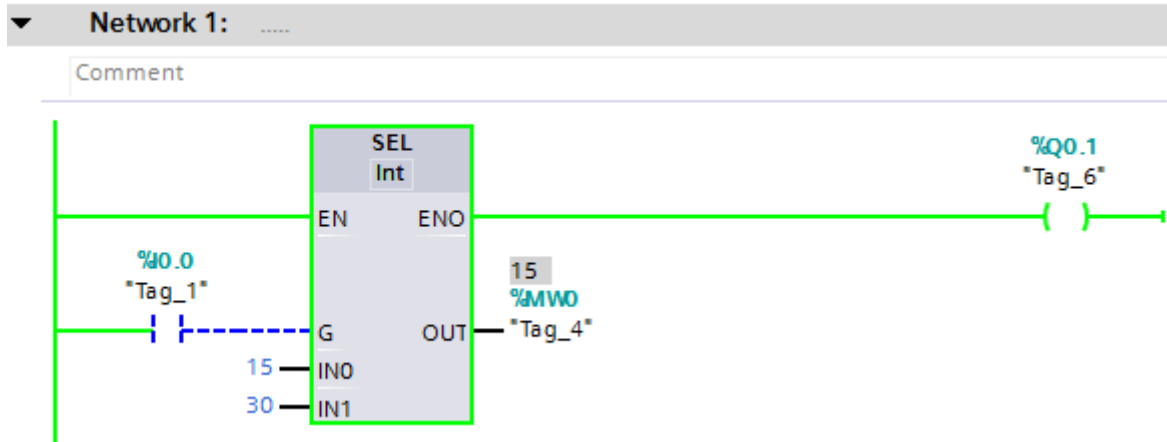
6-İKİLİ GİRİŞTEN SEÇME (SEL) FONKSİYONU

"G" girişine uygulanan sinyale bağlı olarak **"IN0"** veya **"IN1"** girişlerine yazılan değerlerden birisi **"OUT"** çıkışına aktarılır. **"G"** giriş sinyali durumu **"0"** olduğunda **"IN0"** girişindeki değer, **"1"** olduğunda, **"IN1"** girişindeki değer **"OUT"** çıkışına yazılır.

"G" girişine **"BOOL"** (ikili) bir sayı tanımlanmalıdır. Bu sayı direkt yazılabileceği gibi (0 veya 1, FALSE veya TRUE) girişe bağlanan bir kontak bilgisi de kullanılabilir.

SAYISAL (WORD) LOJİK İŞLEMLERİ

6-İKİLİ GİRİŞTEN SEÇME (SEL) FONKSİYONU



SAYISAL (WORD) LOJİK İŞLEMLERİ

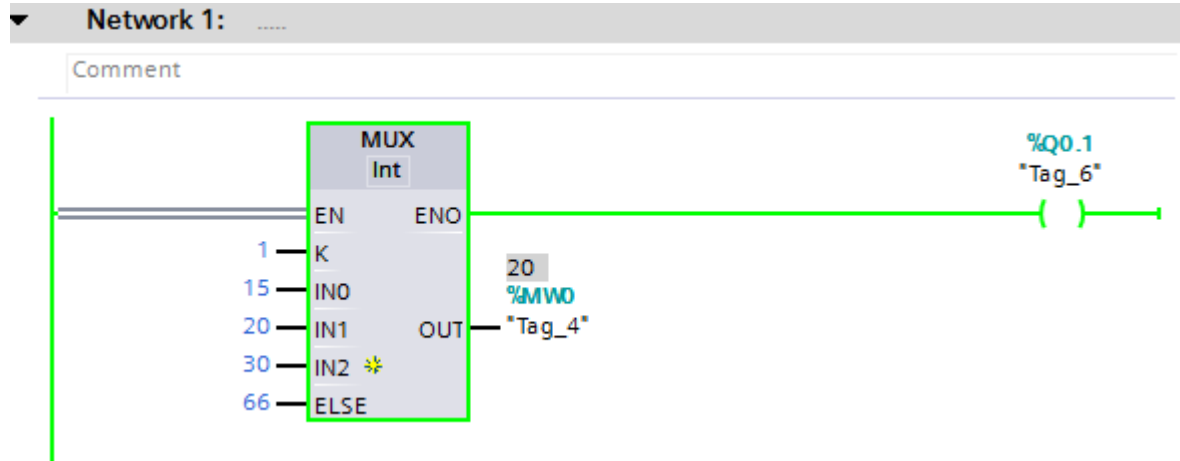
7-ÇOKLU GİRİŞTEN SEÇME (MUX) FONKSİYONU

Girişlerine uygulanan çok sayıdaki sayı ve bellek alanı içeriklerinden, istenen girişteki sayı veya bellek alanının içeriğini çıkışa gönderir.

“K” girişinde bir sayı tanımlanmalıdır. Bu sayı direkt yazılabileceği gibi **“UINT,BYTE,WORD,DATE”** gibi bir bellek alanı içerisindeki sayıda kullanılabilir.

SAYISAL (WORD) LOJİK İŞLEMLERİ

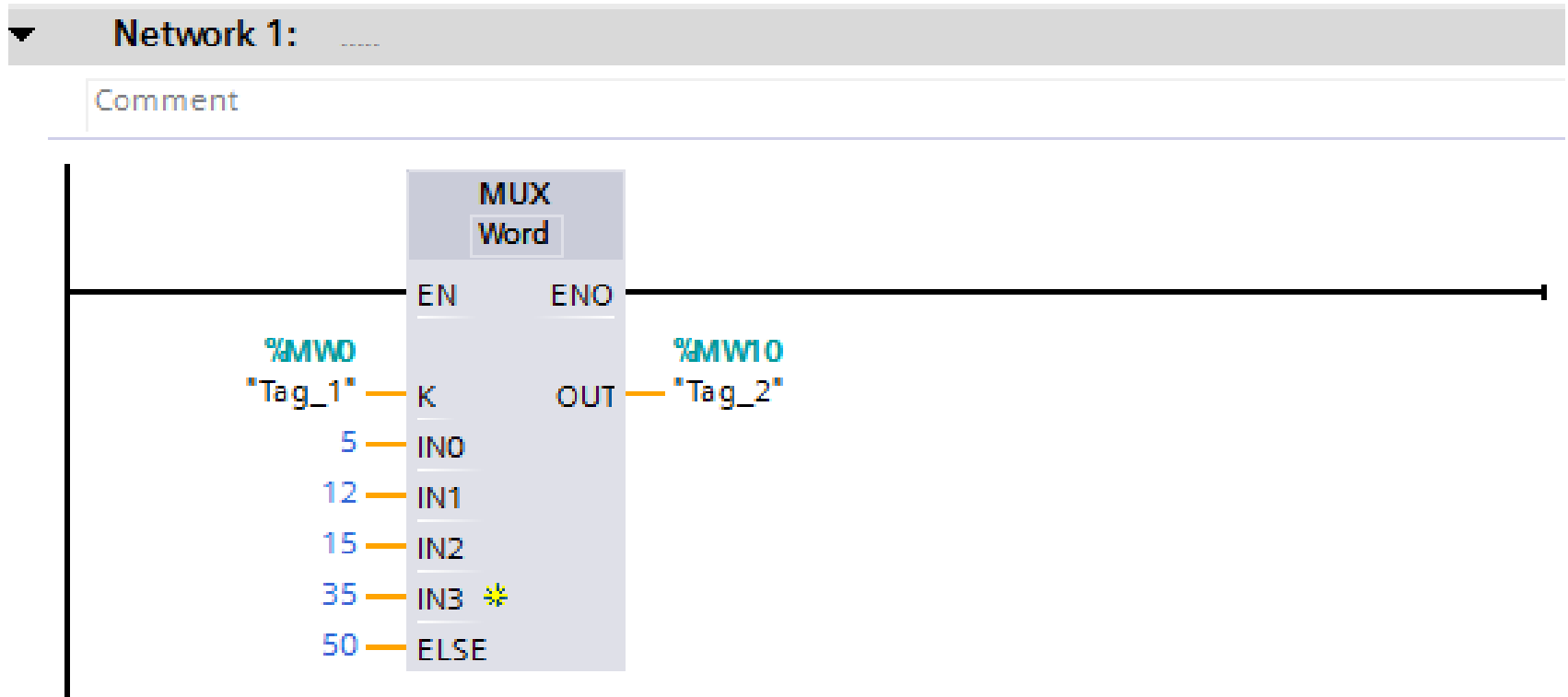
7-ÇOKLU GİRİŞTEN SEÇME (MUX) FONKSİYONU



"K" girişi **"1"** olduğu için **"IN1"**deki değer **"OUT"**a yazılır.

"K"dan girilen sayı değeri numaralı **"IN"** girişi yoksa, **"ELSE"** girişindeki değer **"OUT"**a yazılır.

7-ÇOKLU GİRİŞTEN SEÇME (MUX) FONKSİYONU



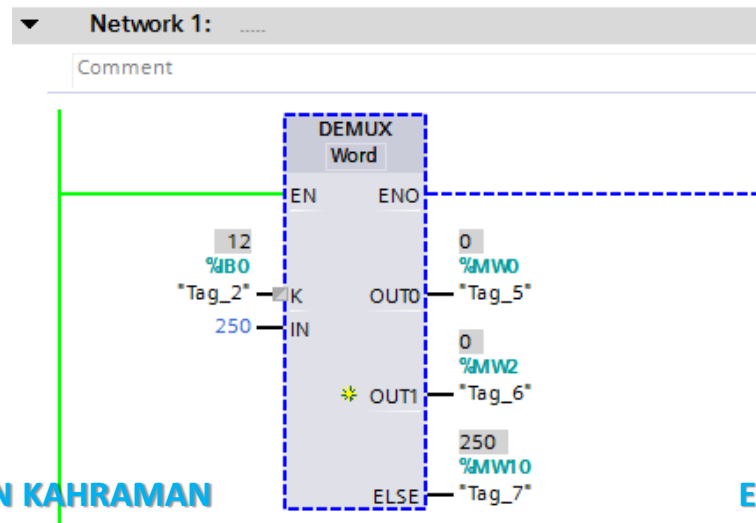
SAYISAL (WORD) LOJİK İŞLEMLERİ

8-GİRİŞİ, SEÇİLEN ÇIKIŞA AKTARMA (DEMUX) FONKSİYONU

"K" girişindeki içeriği **"K"** girişine verilen numaralı çıkışa aktarır.

"K" girişinde bir sayı tanımlanmalıdır. Bu sayı direkt yazılabileceği gibi bir bellek alanı içerisindeki sayıda kullanılabilir.

"K" da girilen sayı değeri kadar **"OUT"** çıkışı tanımlanmamışsa **"ELSE"** çıkışına gönderir.



8-GİRİŞİ, SEÇİLEN ÇIKIŞA AKTARMA (DEMUX) FONKSİYONU

