

- معرفی فرمت اعداد صحیح علامت دار ۱۶ بیتی و ۳۲ بیتی،
- معرفی فرمت اعداد حقیقی
- معرفی فرمت داده ها بصورت باینری و یا هگزا دسیمال
- بررسی مقایسه کننده های برای اعداد صحیح علامت دار و حقیقی
- بررسی توابع ریاضی برای اعداد صحیح علامت دار و اعداد حقیقی
- بررسی توابع مثلثاتی
- بررسی توابع منطقی

۵-۱) فرمت داده ها برای توابع ریاضی و منطقی

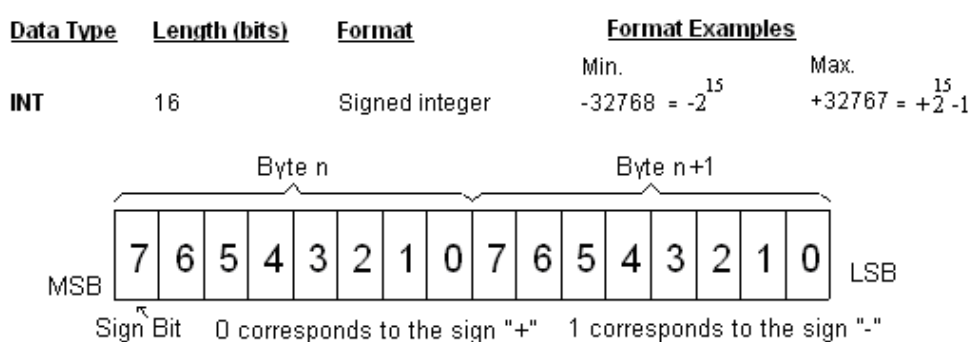
مقدمه

طول داده توابع ریاضی و منطقی در PLC که با اعداد صحیح علامت دار و یا بدون علامت کار می کنند ۱۶ بیتی و یا ۳۲ بیتی است. اعداد صحیح علامت دار ۱۶ بیتی بنام integer و ۳۲ بیتی آن بنام Double Integer نامیده می شوند. توابعی که با این نوع اعداد کار می کنند، مقادیر ثابت در ورودی خود را با فرمت اعداد صحیح علامت دار دسیمال می پذیرند. همچنین داده های بدون علامت ۱۶ بیتی بنام Word و ۳۲ بیتی آن بنام Double Word نامیده می شوند. توابعی که با این نوع داده ها کار می کنند، مقادیر ثابت را در ورودی خود، با فرمت هگزا دسیمال و یا باینری می پذیرند.

در PLC، طول داده برای توابعی که با اعداد حقیقی^۱ کار می کنند ۳۲ بیتی است. فرمت این داده بشکل ممیز شناور^۲ و منطبق با استاندارد IEEE FP 32 است. در ادامه فرمت هر یک از داده ها ذکر شده در بالا معرفی می شوند.

۵-۱-۱) معرفی مجموعه اعداد صحیح علامت دار ۱۶ بیتی (Integer)

در شکل زیر فرمت اعداد صحیح علامت دار ۱۶ بیتی نشان داده شده که پر ارزش ترین بیت آن بیت علامت و ۱۵ بیت دیگر مقدار ارزش اعداد را مشخص می کند. اگر بیت علامت یک باشد محتوای ۱۵ بیت دیگر بصورت کد مکمل^۳ باینری مشخص کننده ارزش عدد صحیح منفی و اگر بیت علامت صفر باشد محتوای ۱۵ بیت دیگر بصورت کد باینری مشخص کننده ارزش عدد صحیح مثبت هستند.



نحوه مقدار دهی ثابت: توابعی که در ورودی خود اعداد Integer می پذیرند برای مقداردهی ثابت، بصورت فرمت و گستره آمده در زیر عمل می شوند.

^۱ Real

^۲ Floating Point

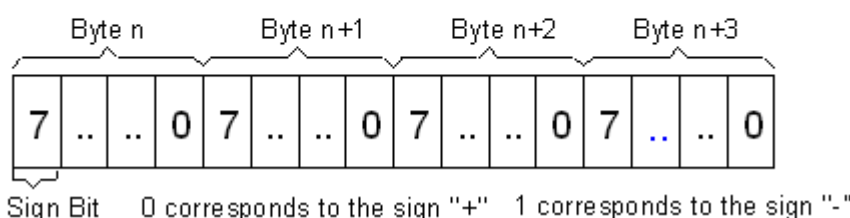
^۳ Two's Complement

فرمت برای مقدار دهی اعداد مثبت ثابت، بصورت $+N$ است که N می تواند صفر تا ۳۲۷۶۷ باشد همچنین برای مقدار دهی اعداد منفی ثابت، فرمت بصورت $-N$ است که N می تواند یک تا ۳۲۷۶۸ باشد.

۵-۱-۲) معرفی مجموعه اعداد صحیح علامت دار ۳۲ بیتی (Double Integer)

در زیر فرمت اعداد صحیح علامت دار ۳۲ بیتی نشان داده شده که با ارزش ترین بیت آن بیت علامت و ۳۱ بیت دیگر مقدار ارزش اعداد را مشخص می کنند. اگر بیت علامت یک باشد محتوای ۳۱ بیت دیگر بصورت کد مکمل ۲ باینری مشخص کننده ارزش عدد صحیح منفی و اگر بیت علامت صفر باشد محتوای ۳۱ بیت دیگر بصورت کد باینری مشخص کننده ارزش عدد صحیح مثبت هستند.

<u>Data Type</u>	<u>Length (bits)</u>	<u>Format</u>	<u>Format Examples</u>	
			Min.	Max.
DINT	32	Signed integer	$L\#-2147483648 = -2^{31}$	$L\#+2147483647 = +2^{31}-1$



نحوه مقدار دهی ثابت: توابعی که در ورودی خود اعداد Double Integer می پذیرند برای مقداردهی ثابت، بصورت فرمت و گستره آمده در زیر مقدار دهی می شوند.

فرمت برای مقدار دهی اعداد مثبت ثابت، بصورت $L\#+N$ است که N می تواند صفر تا ۲۱۴۷۷۸۳۶۷ باشد همچنین برای مقدار دهی اعداد منفی ثابت، فرمت بصورت $L\#-N$ است که N می تواند یک تا ۲۱۴۷۷۸۳۶۸ باشد.

۵-۱-۳) معرفی مجموعه داده بدون علامت ۱۶ بیتی

فرمت داده های بدون علامت ۱۶ بیتی در زیر نشان داده شده است. توابعی که در ورودی خود این نوع داده ها را می پذیرند، برای مقدار دهی ثابت در ورودی آنها می توان بصورت باینری، هگزا دیسیمال و یا بایت ثابت، بصورت فرمت های نشان داده شده در مثال های زیر مقدار دهی کرد.

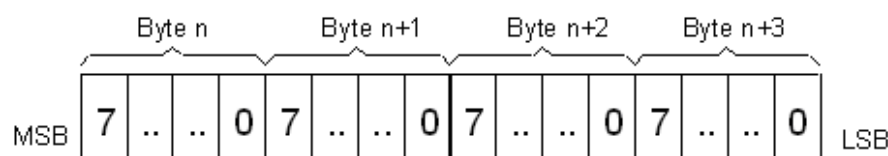
<u>Data Type</u>	<u>Length (bits)</u>	<u>Format</u>	<u>Format Examples</u>	
			Min.	Max.
WORD	16	Binary	2#0	2#1111111111111111
		Hexadecimal	W#16#0	W#16#FFFF
		Unsigned bytes	B#(0,0)	B#(255,255)

توضیح: توابعی که در ورودی خود داده‌های بدون علامت، ۱۶ بیتی (Word) می‌پذیرند، علاوه بر فرمت‌های آمده در بالا می‌توانند در ورودی خود داده بطول بایت را هم بصورت فرمت B#16#nn بپذیرند که در این فرمت nn یک عدد هگزا دسیمال دورقمی است.

۵-۱-۴) معرفی مجموعه داده بدون علامت ۳۲ بیتی

فرمت داده‌های بدون علامت ۳۲ بیتی در زیر نشان داده شده است. توابعی که در ورودی خود این نوع داده‌ها را می‌پذیرند، برای مقدار دهی ثابت در ورودی آنها می‌توان بصورت باینری، هگزا دسیمال و یا بایت بدون علامت، به شکل فرمت‌های نشان داده شده در مثال‌های زیر مقدار دهی کرد.

Data Type	Length (bits)	Format	Format Examples
DWORD	32	Binary, min.	2#00000000000000000000000000000000
		Binary, max.	2#11111111111111111111111111111111
		Min.	Max.
		Hexadecimal	DW#16#0 DW#16#FFFFFFFF
		Unsigned bytes	B#(0,0,0,0) B#(255,255,255,255)

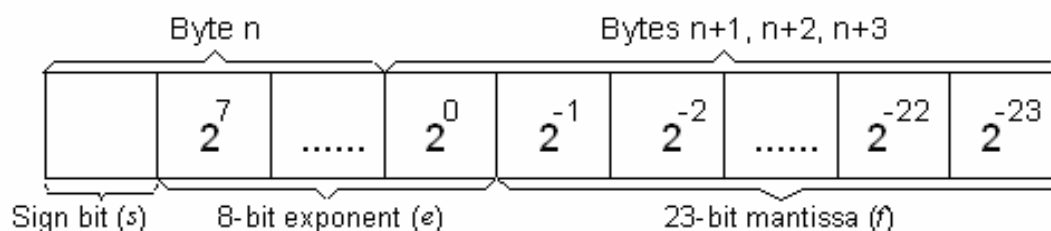


توضیح: توابعی که در ورودی خود داده‌های بدون علامت، ۳۲ بیتی (DWord) می‌پذیرند، می‌توانند در ورودی خود داده ثابت بطول بایت را با فرمت B#16#nn و یا داده ثابت بطول Word را با فرمت W#16#nnnn بپذیرند که در این فرمت‌ها هر یک از nnها یک عدد هگزا دسیمال یک رقمی می‌باشند.

۵-۱-۵) معرفی مجموعه اعداد حقیقی^۱

در PLCهای شرکت زیمنس اعداد حقیقی با فرمت ممیز شناور (با استاندارد IEEE FP 32) که ۳۲ بیتی است، استفاده می‌شوند. فرمت این اعداد در زیر نشان داده شده است.

S7 CPU memory format for the floating-point number (32 bits)



بطوریکه از روی فرمت این اعداد ۳۲ بیتی مشاهده می‌شود، با ارزش‌ترین بیت آن مختص بیت علامت (s)، ۸ بیت مجاور بیت علامت برای نما (e) و ۲۳ بت باقی مانده برای مانتیس (f) است که ارزش هر یک از این بیت‌ها در این فرمت مشخص شده است.

^۱ Real

بر طبق استاندارد IEEE FP 32، استفاده از مقادیر بیت علامت، نما و مانتیس در رابطه زیر، ارزش اعداد ممیز شناور مورد استفاده در S7 CPU تعیین می‌شود. در ادامه توضیحات دقیق‌تری از این متغیرها و رابطه تعیین کننده ارزش اعداد شناور ارائه شده است.

توجه شود در این استاندارد فاصله دو عدد متوالی برای اعداد کوچک، کوچک‌تر و برای اعداد بزرگ، بزرگ‌تر می‌شود. این مطلب را با بررسی رابطه تعیین کننده ارزش اعداد شناور که در زیر آمده است می‌توان برداشت کرد.

S7 CPUs use the IEEE FP 32-bit binary memory format to process floating point numbers:

$$\text{Any floating-point number} = (-1^s)(1.f)(2^{e-127})$$

Where in the normalized number equation shown above:

s = sign bit, (0 = positive, 1 = negative)

e = a binary integer exponent = the decimal integer exponent + 127 as an 8-bit unsigned binary integer exponent ($0 < e < 255$)

f = a 23-bit mantissa with the MSB equal to 2^{-1} and LSB equal to 2^{-23} .

در زیر گستره دامنه اعداد مثبت و منفی در این استاندارد نشان داده شده است.

<u>Data Type</u>	<u>Length (bits)</u>	<u>Format</u>	<u>Format Examples</u>	
REAL	32	Floating point	positive min.	positive max
			+1.175495e-38	+3.402823e+38
			negative min.	negative max.
			-1.175495e-38	-3.402823e+38

نحوه مقدار دهی ثابت: توابعی که در ورودی خود اعداد را بصورت اعداد حقیقی دریافت می‌کنند، برای مقدار دهی ثابت بدو صورت زیر عمل می‌شود.

صورت کسری: $\pm N.M$

که در آن N مقدار عدد صحیح و M بخش کسری عدد است. اگر هر یک از اعداد M و N وجود نداشته باشند بجای آنها صفر منظور می‌شود.

صورت نمائی: $\pm Ne \pm P$

که در آن N مقدار عدد صحیح و P بخش نمای عدد 10 است. اگر در این عدد بخش کسری وجود نداشته باشد بجای P صفر منظور می‌شود.

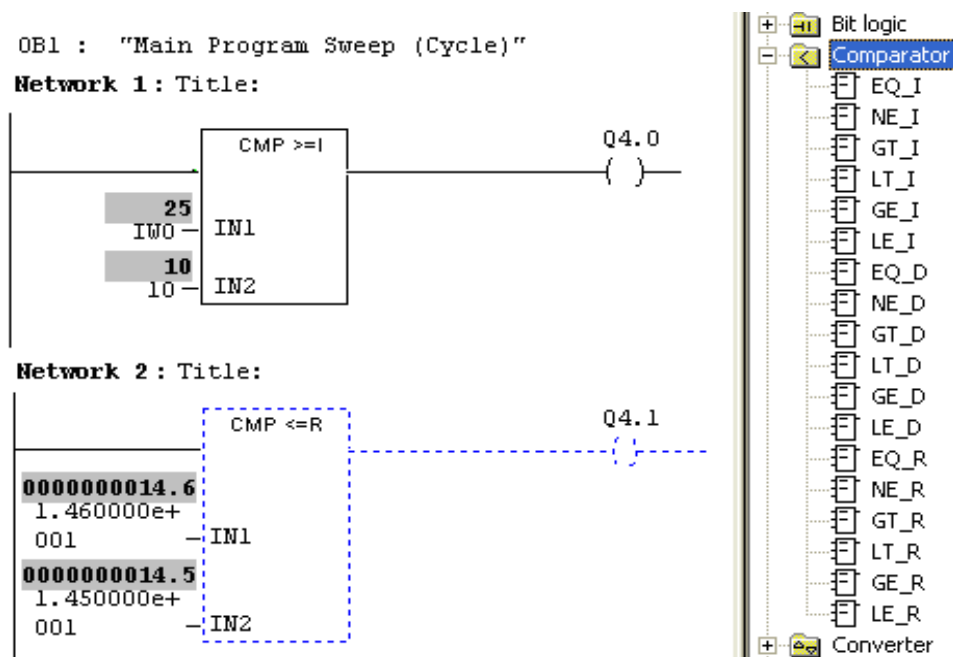
۲-۵ بررسی توابع مقایسه (Comparator)

مجموعه توابع عملیات مقایسه برای اعداد صحیح علامت‌دار (Integer و Double Integer) و اعداد حقیقی (Real) در سمت راست شکل (۱-۵) آمده است.

این توابع از بالا به پایین شامل مقایسه دو عدد صحیح علامت‌دار ۱۶ بیتی (Integer) برای حالات: مساوی، نامساوی، بزرگتر، کوچکتر، بزرگتر و یا مساوی، کوچکتر و یا مساوی آمده است، در ادامه این مجموعه توابع دو مجموعه توابعی دیگر برای مقایسه دو عدد صحیح علامت‌دار ۳۲ بیتی (Double Integer) و برای مقایسه دو عدد حقیقی (نظیر توابع نامبرده شده در بالا) آمده است. این دو مجموعه توابع مشابه مجموعه توابع اعداد Integer هستند.

در سمت چپ شکل (۱-۵) یک تمرین با دو مثال در دو شبکه ارائه شده است که در شبکه اول عمل مقایسه دو عدد صحیح علامت‌دار ۱۶ بیتی (Integer) برای تعیین بزرگتر و یا مساوی انجام می‌شود. کار این دو مثال بررسی شود.

توجه شود، در این نوع توابع، ورودی IN1 با ورودی IN2 مقایسه شده و نتیجه بصورت یک بیت در خروجی ظاهر می‌شود.



شکل (۱-۵)

تمرین: برنامه‌ای آماده کنید تا ورودی IW0 را با اعداد 20 و 10 مقایسه کرده بطوریکه اگر ورودی کوچکتر از 10 بود خروجی Q4.0 یک و اگر ورودی بزرگتر از 20 بود این خروجی صفر شود و اگر ورودی بین دو عدد بود خروجی تغییر نکند.

۳-۵ بررسی توابع عملیات ریاضی عدد صحیح (Integer Function)

مجموعه توابع عملیات ریاضی برای اعداد صحیح علامت‌دار در سمت راست شکل (۲-۵) آمده است. این توابع از بالا به پایین شامل جمع، تفریق، ضرب، تقسیم برای اعداد با طول ۱۶ بیتی (Integer) و در ادامه توابع جمع، تفریق، ضرب، تقسیم و تعیین باقی‌مانده برای اعداد با طول ۳۲ بیتی (Double Integer) می‌باشند.

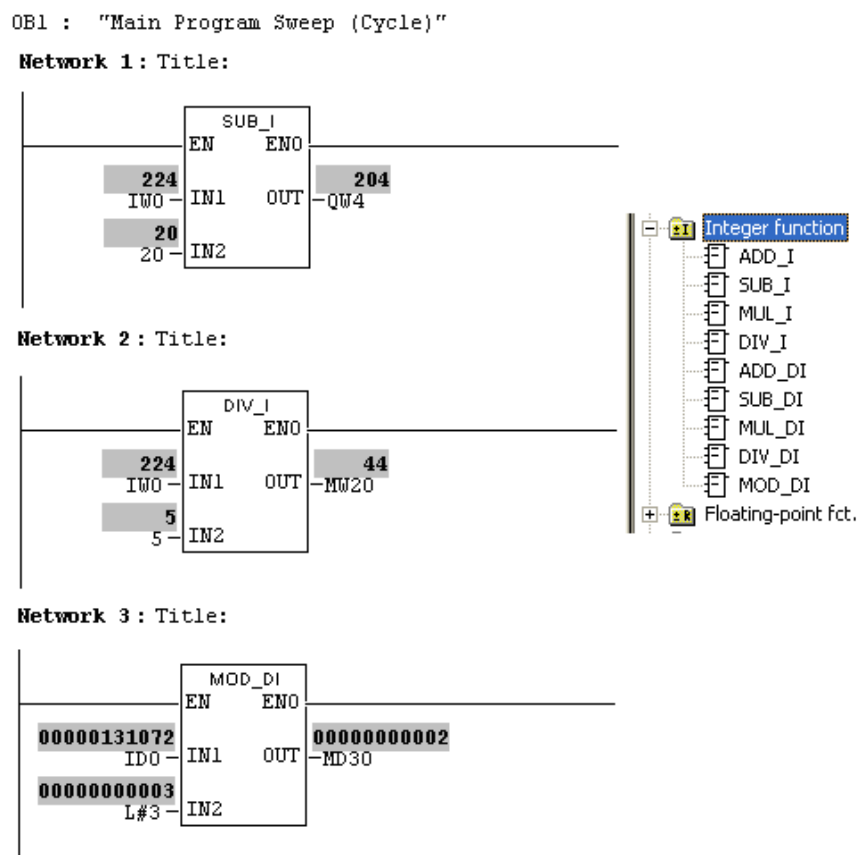
مثال: در سمت چپ شکل زیر یک مثال برای سه تابع ریاضی ارائه شده است که:

در شبکه اول تفاضل IN2 از IN1

در شبکه دوم خارج قسمت عمل تقسیم IN1 بر IN2

در شبکه سوم باقی‌مانده عمل تقسیم IN1 بر IN2 حاصل می‌شود.

توضیح: در این بلوک‌ها اگر نتیجه خارج از گستره قابل قبول قرار گیرد خروجی EN0 صفر می‌شود. این موضوع را در یکی از بلوک‌های تفریق و یا تقسیم بررسی کنید. برای این بررسی لازم است از EN0 استفاده شود. مثلاً نتیجه EN0 به یک بیت از حافظه منتقل شود.



شکل (۵-۲)

تمرین، برنامه‌ای آماده کنید تا مقادیر ورودی IB0 را بر IB1 تقسیم کرده، خارج قسمت این تقسیم را به خروجی QB4 و باقی‌مانده آنرا به خروجی QB5 منتقل کند. مقادیر دو ورودی IB0 و IB1 دو عدد مثبت هستند. وضعیت خروجی EN0 تقسیم‌کننده را به یک بیت حافظه منتقل کرده، و بررسی شود در شرایطی که عمل تقسیم بر صفر انجام می‌شود. وضعیت این بیت چگونه است؟

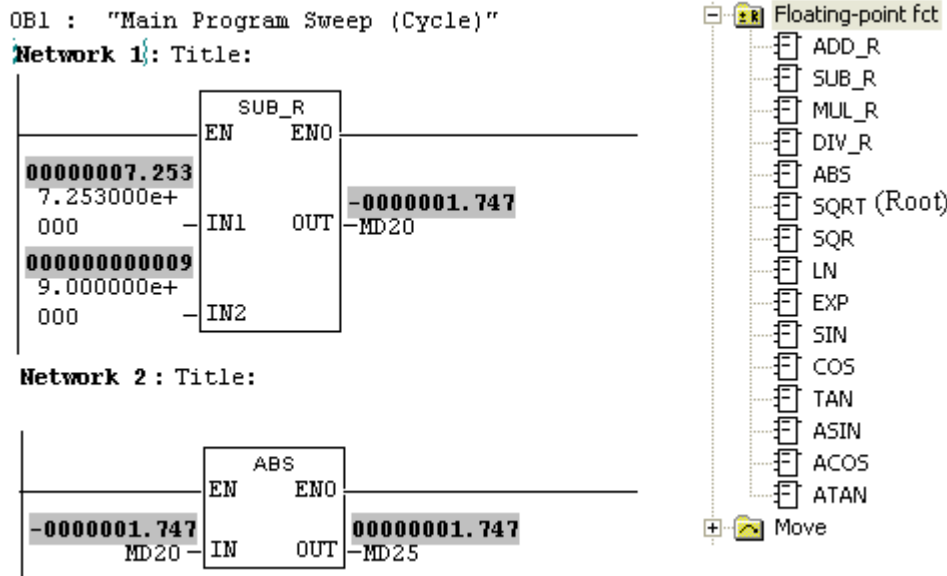
یادآوری: داده‌ها در ورودی‌های IN1 و IN2 این توابع بصورت Integer و یا Double Integer است. اگر در ورود تابع Integer مقدار ثابت لازم باشد، این مقدار بصورت عدد دسیمال علامت‌دار و همچنین اگر در ورودی تابع Double Integer مقدار ثابت لازم باشد، این مقدار بصورت L#n (که n عدد دسیمال علامت‌دار است) تعیین می‌شود.

۴-۵) توابع عملیات ریاضی عدد حقیقی (floating Point Function)

مجموعه توابع عملیات ریاضی برای اعداد حقیقی در سمت راست شکل (۵-۳) آمده است. این توابع از بالا به پائین شامل جمع، تفریق، ضرب، تقسیم، قدرمطلق، جذر، مربع، LN، EXP و در ادامه توابع مثلثاتی و عکس توابع مثلثاتی این اعداد می‌باشند.

توجه شود، واحد زوایا در توابع مثلثاتی رادیان است.

مثال: در سمت چپ شکل (۵-۳) یک مثال برای بررسی دو تابع ریاضی ارائه شده است. در این مثال دو عدد حقیقی ثابت از هم کم و از نتیجه آن قدرمطلق گرفته شده است.



شکل (۵-۳)

تمرین ۱: برنامه‌ای آماده کنید تا آن برنامه از عدد یک سینوس معکوس گرفته، نتیجه را در عدد ۲ ضرب و از حاصل آن کسینوس بگیرد.

تمرین ۲: برنامه‌ای طراحی کنید تا آن برنامه دو عدد Integer به نام‌های IN1 و IN2 را از دو ورودی PLC دریافت کرده، سپس ورودی IN1 را به IN2 رسانده نتیجه را بخروجی PLC با فرمت Double Integer منتقل کند

۵-۵) توابع عملیات جابجائی و چرخشی (Shift و Rotate)

مجموعه توابع جابجائی و چرخشی برای اعداد صحیح علامت‌دار و داده بدون علامت ۱۶ و ۳۲ بیتی در شکل (۵-۴) آمده است. این توابع از بالا به پایین شامل جابجائی به سمت راست برای اعداد Integer و Double Integer (اعداد صحیح علامت دار ۱۶ و ۳۲)، جابجائی به سمت راست و چپ برای داده بدون علامت ۱۶ بیتی (Word) و در ادامه جابجائی به سمت راست و چپ برای داده بدون علامت ۳۲ بیتی (Double Word) و در نهایت چرخش به سمت چپ و راست برای داده بدون علامت ۳۲ بیتی آمده است. در همه این توابع، ورودی IN مربوطه به عددی است که جابجائی یا چرخش باید روی آن انجام شود، و ورودی N مربوطه به عددی است که مقدار آن تعداد دفعات جابجائی یا چرخش برای عدد IN را تعیین می‌کند.

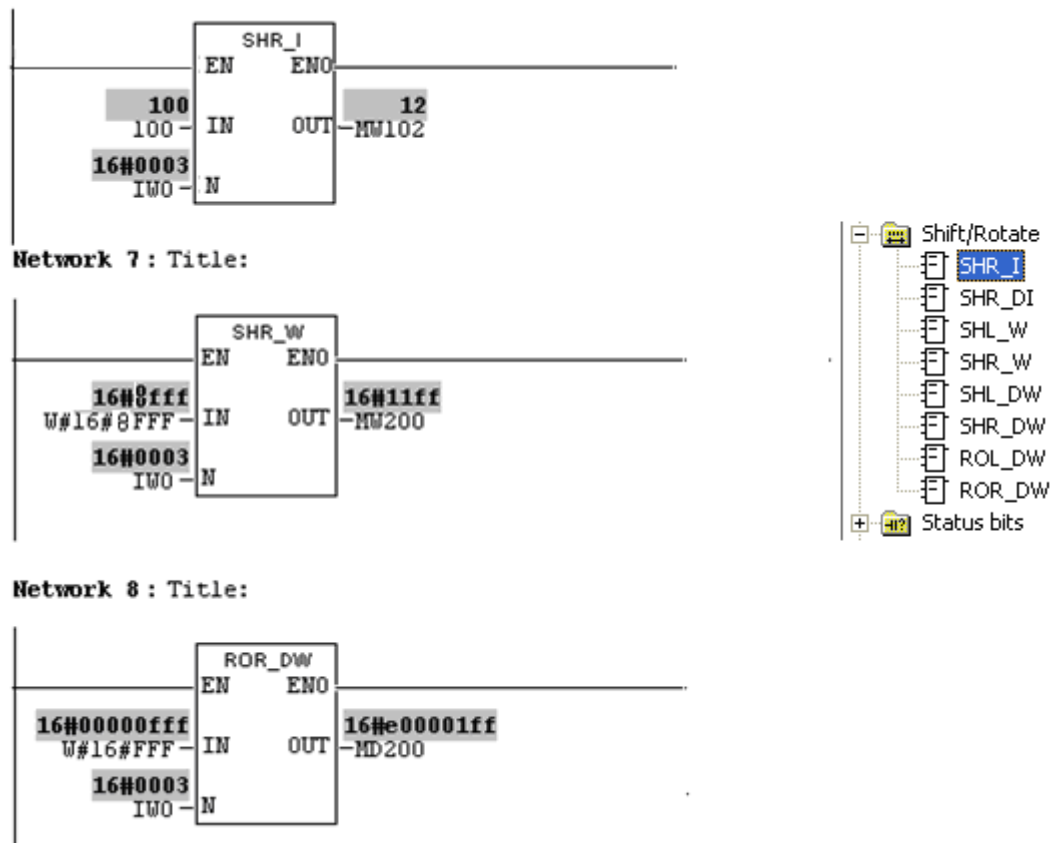
باید توجه شود در همه این توابع فرمت ورودی N بصورت هگزا دسیمال ۱۶ بیتی (Word) بوده و اگر لازم باشد مقدار این ورودی ثابت تعیین شود، فرمت آن بصورت W#16#..... است.

مثال: در سمت چپ شکل (۵-۴) یک مثال برای آشنائی با توابع جابجائی و چرخشی در سه شبکه ارائه شده است - در شبکه اول عدد صحیح علامت دار 100 دسیمال (که معادل باینری IN= 0000 0000 0110 0100 است) به تعداد N=3 مرتبه جابجا و نتیجه آن که معادل عدد 12 دسیمال (که معادل باینری آن 0000 0000 0000 1100 است) به خروجی منتقل شده است.

توجه شود در عمل شیفت به راست اگر عدد صحیح علامت‌دار مثبت باشد با هر مرتبه جابجائی به راست، **صفر** و اگر منفی باشد **یک** از سمت چپ وارد داده ۱۶ (یا ۳۲) بیتی می‌شود. یا به عبارت دیگر با هر بار جابجای عدد علامت دار مثبت و یا منفی، تقسیم بر دو می‌شود. برای عدد مثبت کمترین مقدار نتیجه صفر و برای عدد منفی این نتیجه 1- خواهد بود.

- در شبکه دوم داده بدون علامت ۱۶ بیتی 8fff هگزا دسیمال (که معادل باینری آن IN= 1000 1111 1111 1111 است) به اندازه N=3 مرتبه جابجا و نتیجه آن که معادل 11ff (0001 0001 1111 1111) است به خروجی منتقل شده است.

- در شبکه سوم عدد داده بدون علامت ۳۲ بیتی 00000fff (IN=0000 0000 0000 0000 1111 1111 1111) به اندازه N=3 مرتبه چرخش به راست کرده و نتیجه آن که معادل E00001ff (1110 0000 0000 0000 0001 1111 1111) است به خروجی منتقل شده است.



شکل (۴-۵)

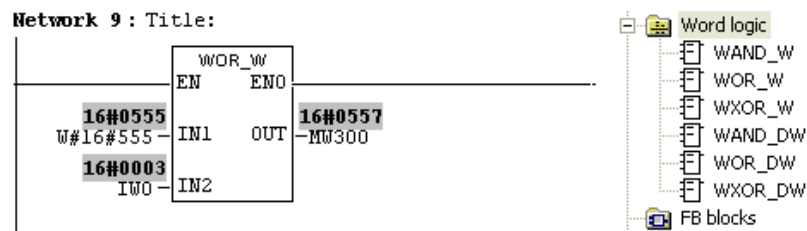
توضیح: چون شیفت به چپ برای Int و Dint مشابه شیفت به چپ Word و Dword هستند، این نوع توابع در نرم افزار نیامده است.

تمرین: شانزده عدد لامپ که بر روی یک دایره نصب شده و به خروجی شانزده بیت QW4 وصل شده‌اند. برنامه ای آمده کنید تا با اجرای آن در هر ثانیه فقط یکی از لامپ‌ها روشن و ترتیب روشن شدن لامپ‌ها در جهت عقربه ساعت باشد.

۶-۵ توابع عملیات Word Logic

مجموعه توابع Word Logic برای داده‌های Word (۱۶ بیتی) و Double Word (۳۲ بیتی) در سمت راست شکل (۵-۵) آمده است. این توابع Logic از بالا به پایین شامل AND, OR, XOR, برای داده‌های ۱۶ بیتی و در ادامه برای داده‌های ۳۲ بیتی هستند.

مثال: در سمت چپ شکل زیر یک مثال برای بررسی یک تابع OR, (۱۶ بیتی) ارائه شده است. در این مثال ورودی IN1 تابع (که دارای عدد ثابت 555 هگزا دسیمال با فرمت W#16#0555 است) با ورودی IN2 تابع (که دارای عدد 0003 هگزا دسیمال دریافت شده از ورودی دیجیتال با آدرس IW0 است) بیت به بیت OR و نتیجه آن (که معادل عدد 0557 هگزا دسیمال و یا 0000 0101 0101 0111 باینری است) از خروجی (Out) تابع به حافظه به طول یک Word (MW300) منتقل شده است.

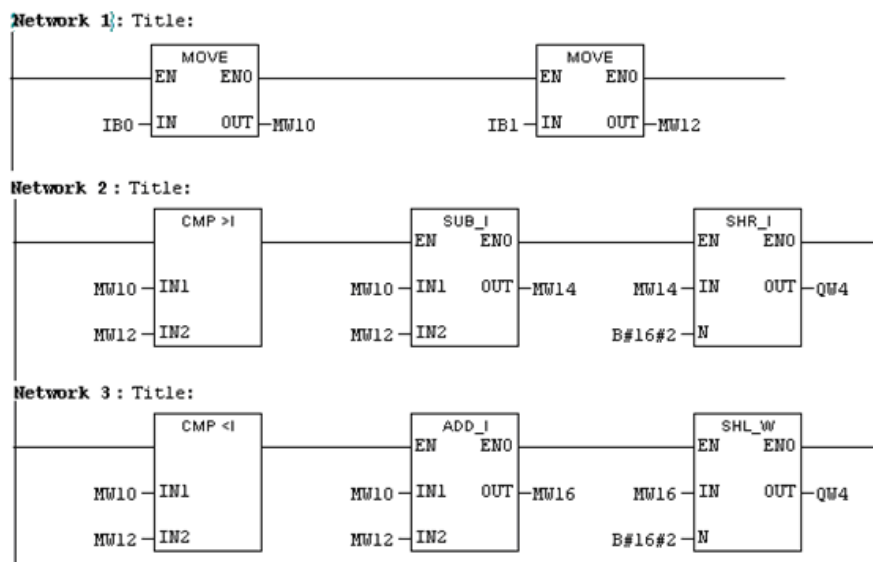


شکل (۵-۵)

تمرین ۱: برنامه‌ای آماده کنید تا دو داده را از ورودی‌های IB0 و IB1 بگیرد و NOR آنها را در خروجی QB4 تحویل دهد.

مثال : برنامه‌ای ایجاد کنید که با آن برنامه، دو عدد A و B با فرمت BYTE را از ورودی‌های IB0 و IB1 دریافت کرده و باهم مقایسه کند، اگر عدد A بزرگتر از عدد B بود حاصل عمل A-B را پس از دو واحد شیفت به راست به خروجی QW4 منتقل کند و اگر A کوچکتر از B بود حاصل عمل A+B را پس از دو واحد شیفت به چپ به خروجی QW4 منتقل کند در صورت مساوی بودن دو عدد A و B، خروجی قبلی تغییر نکند.

پاسخ:



پاسخ تمرین بخش ۵-۲

OB1 : "Main Program Sweep (Cycle)"

