جلسه ششم

هدف:

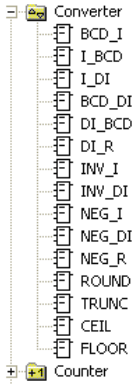
- **بررسی مبدل‌های مختلف فرمت اعداد در PLC شامل: فرمت integer به فرمت کد BCD و بلعکس، فرمت Integer به فرمت Double Integer ، فرمت Double Integer به فرمت Real و بلعکس، متمم ۱ اعداد[۱] integer و Double Integer، متمم دو[۲] اعداد integer و Double Integer، منفی کردن فرمت Real،**
- **بررسی بیت های وضعیت در عملیات CPU شامل،بیت OV[۳]، OS[۴]، BR[۵]، UO[۶]**
- **بررسی بیت‌های وضعیت مربوط به نتایج عملیات ریاضی و منطقی نسبت به صفر**

۶-۱) مبدل‌ها[۷]

مقدمه، مجموعه توابع عملیات تبدیل فرمت‌های مختلف اعداد در شکل (۶-۱) نشان داده شده است. این مبدل‌ها از بالا به پائین شامل:

مبدل عدد BCD سه رقمی علامت‌دار به عدد صحیح علامت‌دار ۱۶ بیتی (Integer)
مبدل عدد صحیح علامت‌دار ۱۶ بیتی (Integer) به عدد BCD سه رقمی علامت‌دار،
مبدل عدد صحیح علامت دار ۱۶ بیتی (Integer) به عدد صحیح علامت دار ۳۲ بیتی (Double Integer)
مبدل عدد BCD هفت رقمی علامت‌دار به عدد صحیح علامت‌دار ۳۲ بیتی (Double Integer)
مبدل عدد صحیح علامت‌دار ۳۲ بیتی (Double Integer) به عدد BCD هفت رقمی علامت‌دار
مبدل عدد صحیح علامت‌دار ۳۲ بیتی (Double Integer) به فرمت عدد حقیقی (Real)
مبدل متمم یک (Once Complement) عدد صحیح علامت دار ۱۶ بیتی (Integer)
مبدل متمم یک (Once Complement) عدد صحیح علامت دار ۳۲ بیتی (Double Integer)
مبدل عدد صحیح علامت دار مثبت ۱۶ بیتی (Integer) به مکمل دو (Twos Complement)
مبدل عدد صحیح علامت دار مثبت ۳۲ بیتی (Integer) به مکمل دو (Twos Complement)
مبدل منفی کردن عددحقیقی (ضرب (۱-) در عدد حقیقی ورودی)
و چهار نوع مبدل فرمت اعداد Real به Double Integer می‌باشند.

در این بخش نحوه عملکرد هر یک از این مبدل‌های با مثالی بررسی می‌شوند.



شکل(۶-۱)

---

[۱] Invert
[۲] Tow's complement
[۳] Over Flow
[۴] Store Over Flow
[۵] Binary Result
[۶] Unordered
[7] Converter

۶-۱-۱) مبدل فرمت BCD به Integer:

طول هر یک از داده ورودی و خروجی این مبدل یک Word، و محدوده اعدادی را که در ورودی خود می‌تواند بپذیرد از ۹۹۹- تا ۹۹۹ است.

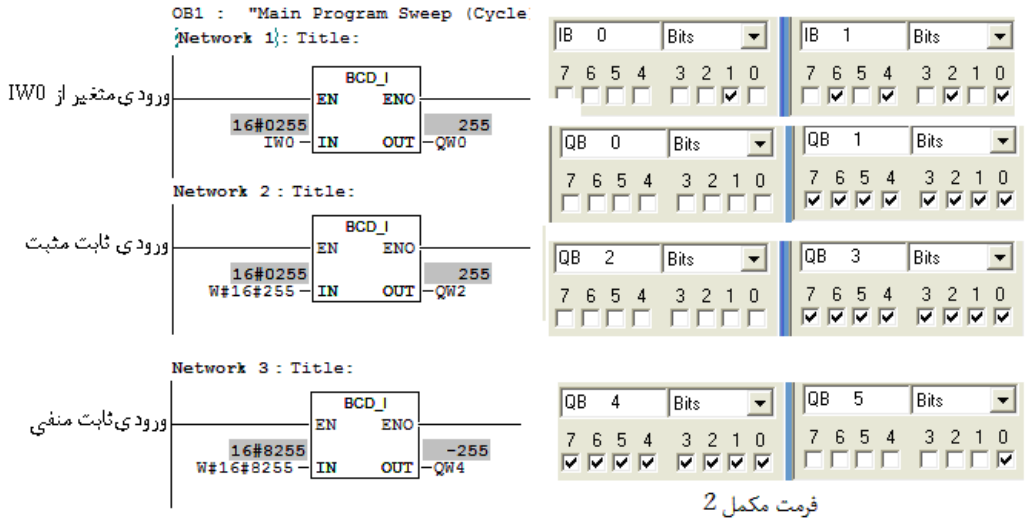
مثال: در سمت چپ شکل (۶-۲) مثالی از مبدل BCD سه رقمی به Integer ارائه شده است در این مثال:

در شبکه اول، ورودی مبدل، عدد مثبت ۲۵۵ (0101 0101 0010 0000) BCD را از آدرس IW0 دریافت کرده است.

در شبکه دوم عدد **مثبت** ۲۵۵ (0101 0101 0010 0000) به صورت مقدار ثابت، با فرمت BCD، (W#16#0255) به ورودی مبدل داده شده است.

در شبکه سوم عدد **منفی** ۲۵۵- (0101 0101 0010 1000) به صورت مقدار ثابت، با فرمت BCD، (W#16#8255) به ورودی مبدل داده شده است.

مثال فوق را می‌توان با نرم افزار PLCSIM بررسی کرد. نتایج هر سه شبکه این مثال در سمت راست شکل(۶-۲) آورده شده است. **توجه شود،** چون ورودی بلوک شبکه سوم عدد BCD منفی (یعنی با ارزش‌ترین بیت آن یک) بوده، در نتیجه فرمت عدد Integer خروجی آن به شکل کد **مکمل دو**[۸] باینری است.



شکل (۶-۲)

توضیح: فرمت عدد ثابت در ورودی تابع BCD_I بصورت W#!6#snnn که nها عدد سه رقمی BCD و s علامت آن که صفر برای عدد مثبت و ۸ برای عدد منفی است.

۶-۱-۲) مبدل فرمت اعداد Integer به BCD:

طول هر یک از داده‌های ورودی و خروجی این مبدل یک Word و محدوده اعدادی را که در ورودی خود می‌تواند بپذیرد از ۹۹۹- تا ۹۹۹ است.

مثال : در سمت چپ شکل (۶-۳) مثالی از مبدل Integer به BCD سه رقمی ارائه شده است در این مثال:

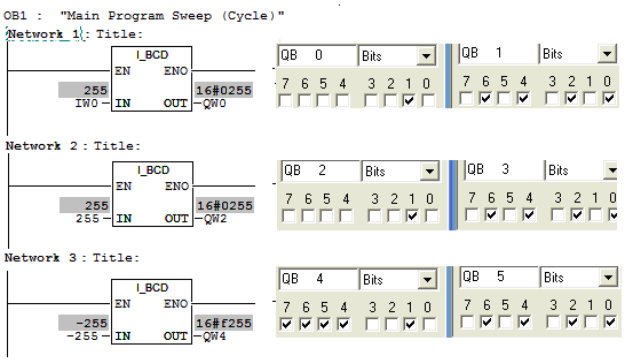
در شبکه اول، ورودی مبدل ، عدد مثبت ۲۵۵ (0000 0000 1111 1111) Integer را از آدرس IW0 دریافت کرده است.

در شبکه دوم عدد **مثبت** ۲۵۵ (0000 0000 1111 1111) به شکل مقدار ثابت، با فرمت **Integer** (255) به ورودی مبدل داده شده است.

در شبکه سوم عدد **منفی** ۲۵۵- (1111 1111 0000 0001) به شکل مقدار ثابت، با فرمت **Integer** (255-) به ورودی مبدل داده شده است. این مثال را می‌توان با نرم افزار PLCSIM بررسی کرد. نتایج هر سه شبکه این مثال در سمت راست شکل(۶-۳)

[۸] Twos Compliment

آورده شده است. **توجه شود،** چون ورودی تابع شبکه سوم عدد منفی Integer(کد مکمل دو) بوده، در نتیجه با ارزشترین چهار بیت خروجی آن یک شده است.



شکل (۶-۳)

در اینجا از بررسی نحوه عملکرد مبدل‌های BCD به Double Integer وDouble Integer به BCD صرف نظر شده است، چون این مبدل‌ها مانند دو مبدل قبل عمل می‌کنند و تفاوت آن‌ها در گستره پذیرش اعداد در ورودی این مبدل‌ها است که این گستره‌ها از ۹۹۹۹۹۹۹- تا ۹۹۹۹۹۹۹+ و طول حافظه‌های ورودی خروجی آن‌ها Double Word می‌باشند.

۶-۱-۳)مبدل فرمت Double Integer به Real

طول هر یک از داده‌های ورودی و خروجی این مبدل یک Double Word است.

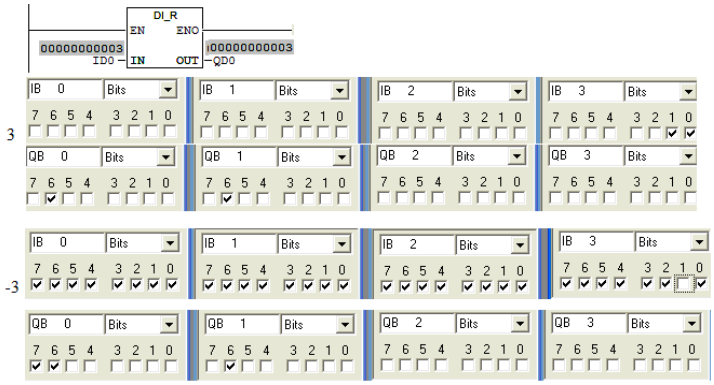
مثال : در شکل (۶-۴) مثالی از مبدل Double Integer به Real ارائه شده است در این مثال داده ورودی مبدل از آدرس ID0 داده دریافت می‌شود:

۱- اگر به ورودی آن عدد مثبت ۳ (که شکل Integer آن بصورت 0011 0000 0000 0000 0000 0000 0000 0000 خواهد بود) داده شود، در حافظه خروجی آن عدد تبدیل شده به فرمت Floating Point به شکل 0000 0000 0000 0000 0100 0100 0000خواهد شد.

۲- اگر به ورودی آن عدد منفی ۳- (که شکل Integer آن بصورت 1101 1111 1111 1111 1111 1111 1111 1111 خواهد بود) داده شود، در حافظه خروجی آن عدد تبدیل شده به فرمت Floating Point، به شکل 0000 0000 0000 0000 0100 1100 0000خواهد شد.

توضیح: این مثال را می‌توان با نرم افزار PLCSIM بررسی کرد. ورودی و خروجی‌های این مبدلها در هر یک از حالت‌های فوق در شکل(۶-۴) نشان داده شده است. **توجه شود،** برای تبدیل اعداد منفی Double Integer به Real فرمت اعداد منفی Double Integer باید بصورت **مکمل دو** باشد.

برای وارد کردن عدد ثابت به ورودی مبدل‌هائی که اعداد Double Intege می‌پذیرند از فرمت L#(+ or -)n استفاده می‌شود، که برای مثال فرمت عدد مثبت ۳ به شکل L#3 و فرمت عدد منفی ۳- به شکل L#-3 است.

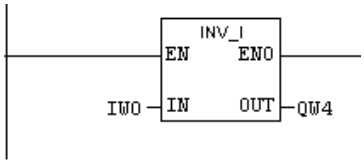


شکل (۶-۴)

۶-۱-۴)مبدل معکوس کردن (Ones Complement) اعداد صحیح علامت‌دار ۱۶ بیتی (Integer)

طول هر یک از داده‌های ورودی و خروجی این مبدل یک Word و نتیجه عمل تبدیل در خروجی آن، NOT بیت-های ورودی (فرمتOnes Complement) است.

برای مثال اگر به ورودی طرح شکل(۶-۵) عدد مثبت ۱۲ با فرمت Integer که شکل باینری آن بصورت 0000 0000 0000 1100 خواهد بود، داده شود، عدد تبدیل شده در حافظه خروجی بصورت 0011 1111 1111 1111 خواهد شد.



شکل(۶-۵)

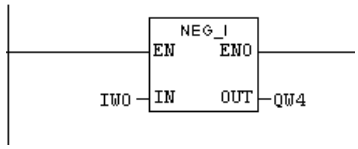
عملکرد معکوس کردن اعداد صحیح علامت دار ۳۲ بیتی (Double Integer) مشابه مبدل معکوس کردن اعداد صحیح علامت‌دار ۱۶ بیت است و تفاوت آنها فقط در طول داده آن‌ها است که اعداد Double Integer ۳۲ بیتی و Integer ۱۶ بیتی می‌باشند. بر این اساس در اینجا از بررسی مثالی از عملکرد معکوس کردن اعداد Double Integer صرف نظر می‌شود.

۶-۱-۵) مبدل مکمل دو(Twos Complement)

طول هریک از داده‌های ورودی و خروجی این مبدل Word و نتیجه عمل تبدیل این مبدل در خروجی، حاصل‌ضرب عدد منفی یک در عدد Integer ورودی است.

برای مثال اگر به ورودی طرح شکل(۶-۶) عدد Integer ۱۲ (که شکل باینری آن بصورت 1100 0000 0000 0000 است) داده شود در حافظه خروجی عدد تبدیل شده بصورت عدد Integer ۱۲- خواهد شد(که معادل باینری آن بصورت 1111 1111 1111 0100 (مکمل دو) است).

و همچنین اگر به ورودی آن عدد Integer ۱۲- (که شکل باینری آن بصورت 0100 1111 1111 1111 (مکمل دو) است) داده شود در حافظه خروجی آن عدد تبدیل شده بصورت عدد Integer ۱۲خواهد شد (که معادل باینری آن بصورت 0000 0000 0000 1100 است).



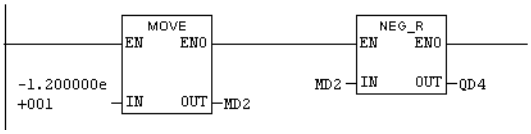
شکل(۶-۶)

عملکرد مکمل دو اعداد صحیح علامت دار ۳۲ بیتی (Double Integer) مشابه مکمل دو اعداد صحیح علامت‌دار ۱۶ بیت است و تفاوت آنها فقط در طول داده آن‌ها است که اعداد Double Integer ۳۲ بیتی و Integer ۱۶ بیتی می‌باشند. بر این اساس در اینجا از بررسی عملکرد مبدل مکمل دو اعداد Double Integer صرف نظر می‌شود.

۶-۱-۶) مبدل عدد Real مثبت به عدد Real منفی (NEG_R)

طول هر یک از داده‌های ورودی و خروجی این مبدل Double Word و نتیجه عمل تبدیل این مبدل در خروجی حاصل‌ضرب عدد 1- در عدد Real ورودی است. در این تبدیل فقط بیت علامت آن تغییر می‌کند.

توضیح: مثال شکل(۶-۷) را می‌توان با سیمولاتور(PLCSIM) بررسی و فرمت حافظه‌های ورودی و خروجی مبدل را مشاهده کرد.



شکل(۶-۷)

در این مثال اگر به ورودی تابع Move عدد 12.0- داده شود فرمت حافظه ورودی تـابع NEG_R 0000 0000 0100 0001 1100 0000 0000 0000 و فرمت حافظه خروجی آن بصورت 0000 0000 0000 0000 0000 0000 0100 0001 0100 0000 خواهد شد.

توضیح: در تابع Move داده با هر فورمت (مثلاINT (n± و یا (n± L#) DINT و یا Real (m.n±)) در ورودی داده شود در خروجی نتایج با همان فورمت انتقال می‌یابد. در مثال شکل(۶-۷) عدد 12.0- در حافظه خروجی تابع Move به شکل 0001 1100 0000 0000 0000 0000 0000 0100 منتقل شده است. و اگر عدد ورودی تابع Move بصورت L#-12 داده شود در خروجی آن نتایج به صورت 0100 1111 1111 1111 1111 1111 1111 11111 1111 منتقل می‌شود.

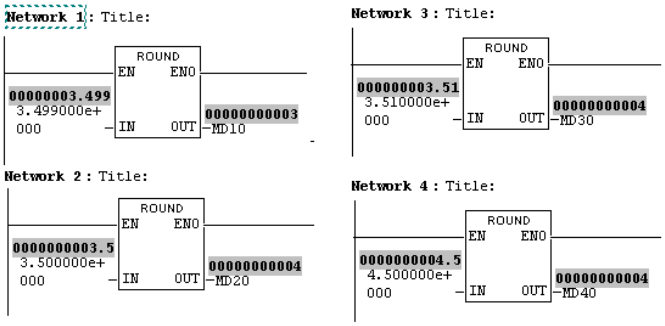
۶-۲)مبدل‌های Real به Double Integer

مقدمه

در این بخش چهار نوع تابع برای تبدیل فرمت Real به فرمت Double Integer بررسی می شود. تفاوت این توابع در روش حذف بخش مقادیر کسری اعداد Real است.. این چهار تابع شامل: ROUND، TRUNC، CEIL و FLOOR هستند که در ادامه هر یک از آنها با یک مثالی بررسی می شوند.

۶-۲-۱) تابع Round:

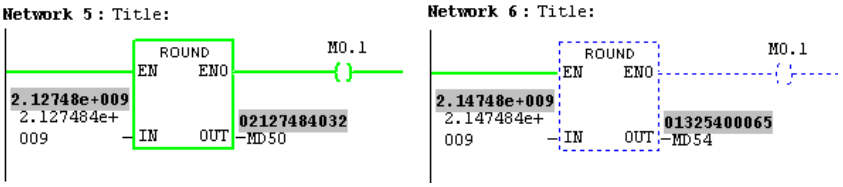
با این تابع عدد حقیقی (با فرمت Floating point) به نزدیک‌ترین عدد صحیح گرد شده و حاصل به یک عدد صحیح علامت دار ۳۲ بیتی(Double Integer) تبدیل می‌شود. اگر عدد حقیقی دقیقا در وسط دو عدد صحیح باشد (مثل N.5) نتیجه بـه عـدد زوج گرد می‌شود. این تابع را می‌توان با طرح شکل(۶-۸) بررسی کرد. بطوریکه از روی شکل مشاهده می‌شود در این مثال عدد ۳/۴۹۹ به عدد ۳ و عدد ۳/۵۱ به۴ و عدد ۴/۵ به ۴ گرد شده‌اند.



شکل(۶-۸)

چون در PLC دامنه تغییرات اعداد حقیقی بزرگتر از دامنه تغییرات عدد Double Integer (عددصحیح علامت دار ۳۲ بیتی) است، اگر حاصل این تبدیل خارج از مجموعه اعداد Double Integer (۲۱۴۷۴۸۳۶۴۷+ ۲۱۴۷۴۸۳۶۴۸-) قرار گیرد، خروجـی سـرریز کرده، و در این وضعیت خروجی ENO تابع صفر می‌شود. این موضوع را می‌توان با طرح شکل(۶-۹) بررسی کرد.

در دو شبکه زیر، به ورودی تابع شبکه پنج عدد ۲۱۴۷۴۸۳۶۴۷/ ۳ و به ورودی شبکه شـش عـدد ۲۱۴۷۴۸۳۶۴۸/ ۳ داده شـده لست، بطوریکه از روی شکل مشاهده می‌شود مقدار خروجی در شبکه شش سرریز(Overflow) کرده و باعث صفر شدن خروجی ENO شده است.



شکل(۶-۹)

۶-۲-۲) تابع TRUNC:

با این تابع، بخش کسری عدد حقیقی با فرمت Floating point به صفر گرد و حاصل به یک عدد Double Integer (عدد صحیح علامت‌دار ۳۲ بیتی) تبدیل می‌شود. برای مثال عدد ۴/۵- به ۴- و ۴/۵+ به ۴+ تبدیل می‌شود.

۶-۲-۳) تابع CEIL :

با این تابع عدد حقیقی با فرمت Floating point به بالا گرد شده و حاصل به یک عدد Double Integer (عدد صحیح علامت‌دار ۳۲ بیتی) تبدیل می‌شود. برای مثال عدد ۴/۵- به ۴- و ۴/۵+ به ۵+ گرد می‌شود.

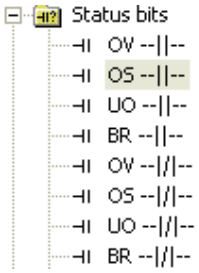
۶-۲-۳) تابع FLOOR:

با این تابع عدد حقیقی با فرمت Floating point به پائین گرد و حاصل به یک عدد Double Integer (عدد صحیح علامت دار ۳۲ بیتی) تبدیل می‌شود. برای مثال عدد ۴/۵- به ۵- و ۴/۵+ به ۴+ گرد می‌شود.

توضیح:همانند تابع Round اگر حاصل هر یک از توابع TRUNK، CELL و FLOOR خارج از مجموعه اعداد Double Integer (۲۱۴۷۴۸۳۶۴۷+ ۲۱۴۷۴۸۳۶۴۸-) قرار گیرد، خروجی سرریز کرده، و در این وضعیت ENO تابع مربوطه صفر می‌شود.

۶-۳ بررسی بیت‌های وضعیت[۱]

با بیت‌های وضعیت آمده در جدول زیر می توان تعدادی از وضعیت عملیات ریاضی و منطقی را تعین کرد. که شرح وضعیت هر یک از آنها در کنار این جداول آمده است.

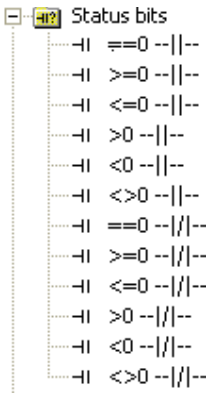


بیت OV[۲]: هرگاه در موقع اجرای یک تابع ریاضی Over Flow رخ دهد (یعنی نتیجه خارج از محدوده اعداد تعریف شده برای آن تابع قرار گیرد) بیت Over Flow فعال (یک) می‌شود. این نتیجه تا اجرای نشدن تابع ریاضی دیگردر برنامه، ثابت باقی‌می‌ماند.

بیت OS[۳]: هرگاه در اجرای یک بلوک Over Flow رخ دهد، بیت OV در بیت OS (Stored OV) ذخیر می‌شود. این بیت ذخیره شده تا خارج شدن اجرای برنامه از آن بلوک (یا پایان دوره اجرای بلوک OB1) در OS باقی می‌ماند.

بیت UO[۴]: هرگاه یکی از متغیرهای یک تابع ریاضی Real خارج از عدد تعریف شده باشد بیت Unordered فعال میشود. این نتیجه تا اجرای تابع ریاضی بعدی در برنامه فعال باقی‌می‌ماند.

بیت BR[۵]: هرگاه یک بلوکی که دارای شرط اجرا برای ورودی EN است، اجرا شود بیت Binary Result فعال می‌شود. این نتیجه تا اجرای بلوک‌بعدی، فعال باقی‌می‌ماند.



بیت‌های وضعیت نتایج عملیات ریاضی و منطقی نسبت به صفر:

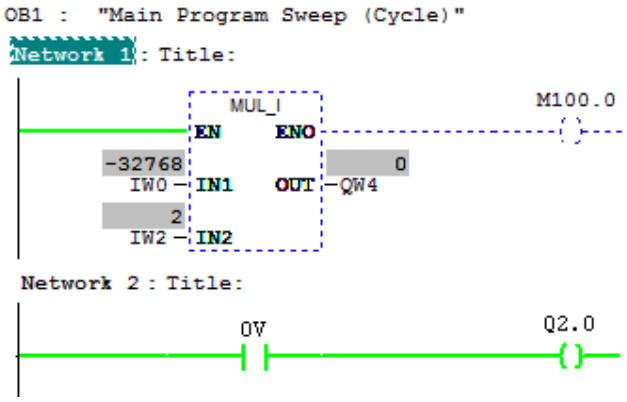
با این بیت‌های وضعیت، می‌توان وضعیت انباره[۶] (رجیسترمحل قرارگیری نتایج عملیات ریاضی و منطقی) را نسبت به صفر بررسی کرد. این بررسی‌ها شامل مساوی، بزرگتر و یا مساوی، کوچکتر و یا مساوی، بزرگتر، کوچکتر و نامساوی صفر است.[۷]

---

[۱] Status Bite
[۲] Over Flow
[۳] Store Over Flow
[۴] Unordered
[۵] Binary Result
[۶] Accumulator

توضیح: در برنامه‌ها می‌توان هر یک از این بیت‌های وضعیت را در دو منطق مختلف (بصورت کنتاکت باز و یا بسته) استفاده کرد.که در ادامه نحوه عملکرد هریک از این بیت‌های وضعیت با یک مثال بررسی می شوند.

۶-۳-۱)مثال برای بررسی بیت OV – هرگاه در طرح شکل(۶-۱۰) در اجرای تابع ریاضی شبکه یک، Over Flow رخ دهد (یعنی نتیجه حاصلضرب خارج از محدوده اعداد ۳۲۷۶۸- تا ۳۲۷۶۷+ واقع شود)، بیت وضعیت OV یک (و خروجی EON تابع ضرب صفر) شده در نتیجه خروجی شبکه دو (Q2.0) یک می شود

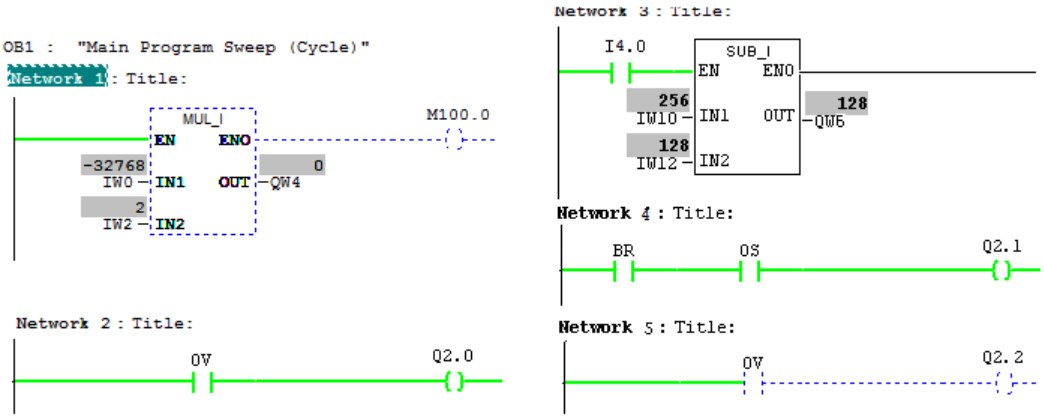


شکل(۶-۱۰)

تمرین: برنامه‌ای آماده کنید تا با آن برنامه، دو عدد Integer را از ورودی‌های IW0 وIW2 دریافت و حاصل ضرب آنها را به خروجی QW4 بصورت Integer منتقل کند و اگر در اجرای این برنامه Over Flow رخ دهد و نتیجه این Over flow مربوط به عدد منفی باشد به خروجی عدد 32000- و اگر مربوط به عدد مثبت باشد به خروجی عدد 32000 منتقل کند.

۶-۳-۲) مثال برای بررسی بیت‌های OS و BR – هرگاه در طرح شکل(۶-۱۱) در شبکه یک Over flow رخ داده و در شبکه سه با فعال شدن I0.0 عمل تفریق به درستی انجام شود، خروجی‌های Q2.0 و Q2.1 فعال می‌شوند.چون بیت OV فعال شده در شبکه یک باعث فعال شدن خروجی Q2.0 در شبک دو می‌شود. همچنین این بیت OV در بیت OS ذخیره شده و با اجرای درست بلوک ریاضی شبکه سه بیت وضعیت BR هم فعال می‌شود و در نتیجه خروجی شبکه چهار که حاصل AND بیت OS و بیت BR است فعال می شود.

هرگاه در شبکه یک عمل ضرب به درستی انجام شود و در شبکه سه با فعال شدن I0.0 در اجرا تابع تفریق Over flow رخ دهد خروجی‌های Q2.1 و Q2.2 فعال خواهند شد. هر دو مورد را بررسی کنید.

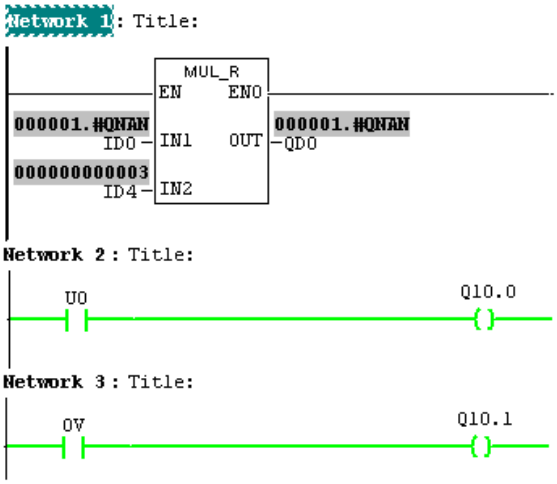


شکل(۶-۱۱)

۶-۳-۳)مثال برای بررسی بیت OU – اگر در طرح شکل(۶-۱۲) به دو ورودی دو عدد

ID0 = 0111 1111 *000 0000 0000 0000 0000 0001
ID4 = 0*11 1111 1000 0000 0000 0000 0000 0000

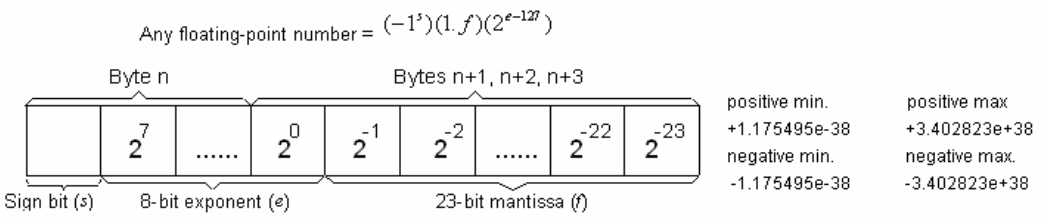
داده شود، به طوریکه بجای دو علامت * (I1.7 و I4.6) مقادیر 0 و یا 1 گذاشته شود. با این تست می‌توان عملکرد بیت‌های UO و OV را در توابع Real بررسی کرد.



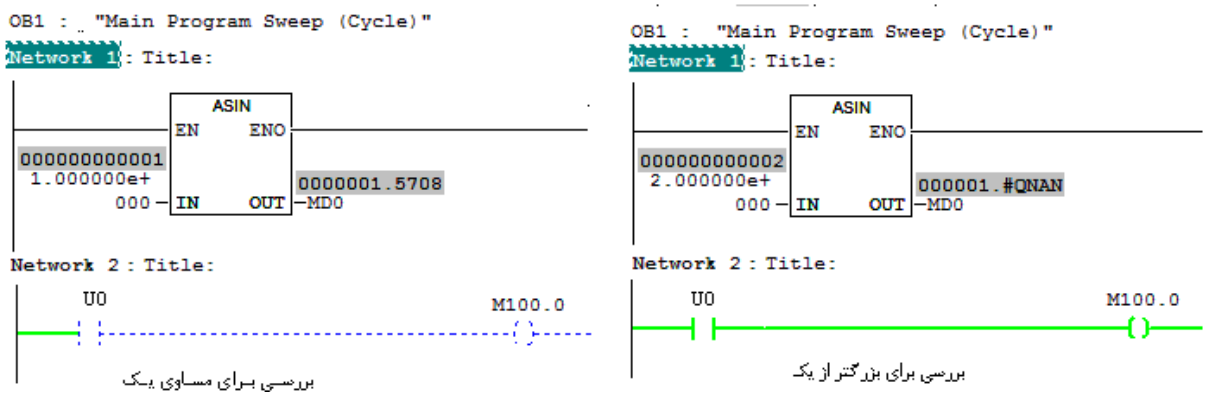
I1.7	I4.6	OV	UO
0	0		
0	1		
1	0		
1	1		

شکل(۶-۱۲)

یاد آوری فرمت اعداد Real

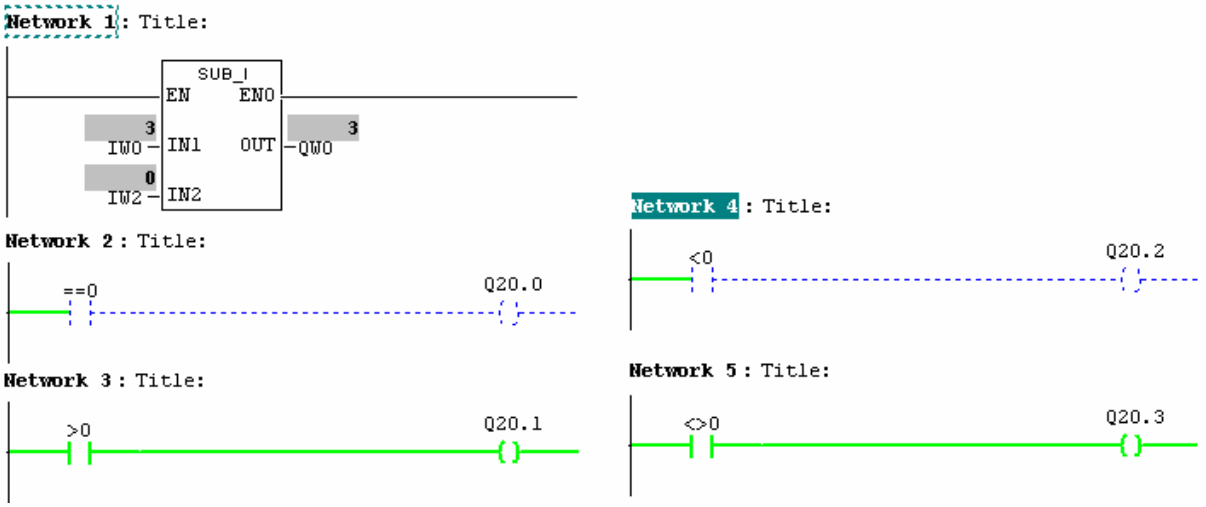


مثال دیگر برای بررسی بیت OU: اگر به ورودی تابع طرح شکل (۶-۱۳)عدد بزرگتر از ۱ داده شود UO فعال می‌شود. این موضوع را بررسی کنید.



شکل(۶-۱۳)

۶-۳-۴) مثال برای بررسی وضعیت Accumulator نسبت به صفر- در CPU یک PLC نتـایج عملیـات **ریاضـی و منطقـی** در درون Accumulator قرار می‌گیرند، برای بررسی وضعیت Accumulator نسبت به صفر، طرح شـکل(۶-۱۴) را بررسـی کـرده، تعـین کنید در چه وضعیتی از Accumulator خروجی‌های Q2.0 تا Q2.3 فعال می‌شوند.

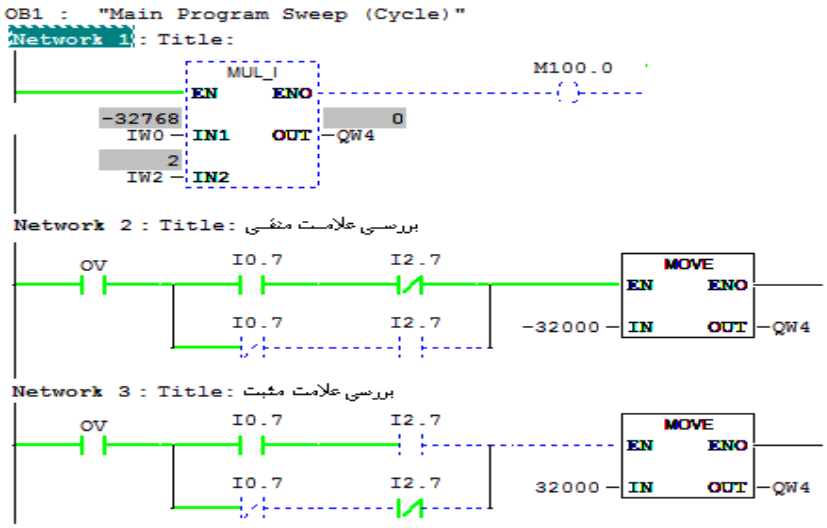


شکل(۶-۱۴)

تمرین : برنامه‌ای طرح کنید که تا در آن برنامه، ورودی IW0 با عدد W#16#00ff بصورت منطقیAND شده، اگر نتیجه غیر صفربود، نتیجه بصورت کد BCD به خروجی QW4 منتقل شود.

- **معرفی ماژول آنالوگ و سامانه[1] حرارتی**

پاسخ تمرین بخش ۶-۳-۱



[1] Plant