INTERMEDIATE PROGRAM

- DIFU & DIFD
- INC & DEC
- Shift Register
- Move & Compare
- Interlocking
DIFU

- Differential UP (rise edge)
- DIFU (13) turns ON the designated bit (B) for one scan on reception of the leading (rising) edge of the input signal.

- Operand Data Areas:
  - B: Bit IO, AR, HR, LR.

©nikmzakmal 2010
DIFD

- Differential Down (fall edge)
- DIFD(14) turns ON the designated bit (B) for one scan on reception of the trailing (falling) edge of the input signal.

- Operand Data Areas:
  - B: Bit IO, AR, HR, LR.
Basic DIFU: exercise 1

Start
Push Start button
Yellow Light ON
End

©nikmzakmal 2008
Basic DIFD: exercise 2
Combination of DIFU/DIFD: exercise 3

Start
- Select Manual Mode
  - Yellow Light ON
- Select Auto Mode
  - Green Light ON

Push Stop Mode
- Yellow & Green Light OFF

End

Program Name: NewProgram1
Section Name: Section1

©nikmzakmal 2010
Move

- Copies the contents of S to D.

```
 MOV(021)  

 S: Source  

 D: Destination
```

```
Source word

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

Destination word

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

Bit status not changed

©nikmzakmal 2010
# MOVE

## Operand Data Areas:

<table>
<thead>
<tr>
<th>S</th>
<th>Source Word</th>
<th>IO, AR, DM, HR, LR, #</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Destination word</td>
<td>IO, AR, DM, HR, LR</td>
</tr>
</tbody>
</table>
Compares (CMP)

- Compares Cp1 and Cp2 and;
- Outputs the result to the GR, EQ and LE flags in the SR area
###Operand Data Areas:

<table>
<thead>
<tr>
<th>cp1</th>
<th>1(^{st}) compare word</th>
<th>IO, AR, DM, HR, TC, LR, #</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp2</td>
<td>2(^{nd}) compare word</td>
<td>IO, AR, DM, HR, TC, LR, #</td>
</tr>
</tbody>
</table>
Basic Move & CMP: exercise 4
Move & CMP: exercise 5

1. **Start**: Push Start Button
2. **Counter 10 X ?**: Yes → Green Light ON, Push Stop Button, Green Light OFF; No → Counter 10 X ?
3. **Yes**: Push Stop Button
4. **End**: Stop

Program 0:

```
0.00  Start
0.01  stop

1. CNT000
    #10

2. MOV(21)
    &A
    140
    confirm counter 10

3. MOV(21)
    #30030
    140
    confirm counter 10
```

Compare: 100.01

Always ON Flag: P_On

Equals (EG) Flag: P_EQ
INCREMENT & DECREMENT

**Purpose:**  
Increments Wd, without affecting Carry (CY).

**Operand Data Areas:**  
Wd : Increment Words

**Purpose:**  
DEC(39) decrements Wd, without affecting CY. DEC(39) works the same way as INC(38) except that it decrements the value instead of incrementing it.

**Operand Data Areas:**  
IO,AR,DM,HR,LR

For INC(38) and DEC(39) the source and result words are the same. That is, the contents of the source word is overwritten with the instruction result.

©nikmzakmal 2008
Basic INC & DEC : exercise 7

- Start
- Select Manual Mode
- Move data #0 to DM00
- Select Auto Mode
- Red Light ON
- Push Start Pb or Stop PB

Compare #5 to DM00
- If Equal=Yellow, Greater=Green, Less=Red

End
SHIFT REGISTER

SFT(10) - Shift Register

Purpose:

SFT(10) is controlled by three execution conditions, I, P and R.

- If SFT(10) is executed and (a) execution condition P is ON and was OFF the last execution and (b) R is OFF, then execution condition I is shifted into the rightmost bit of a shift register defined between St and E, that is, if I is ON, a 1 is shifted into the register; if I is OFF, a 0 is shifted in. When I is shifted into the register, all bits previously in the register are shifted to the left and the leftmost bit of the register is lost.

- The execution condition on P functions like a differentiated instruction, that is, I will be shifted into the register only when P is ON and was OFF the last time SFT(10) was executed. If execution condition P has not changed or has gone from ON to OFF, the shift register will remain unaffected.

- St designates the rightmost word of the shift register; E designates the leftmost. The shift register includes both of these words and all words between them. The same word may be designated for St and E to create a 16-bit (that is, 1-word) shift register.

- When execution condition R goes ON, all bits in the shift register will be turned OFF (that is, set to 0) and the shift register will not operate until R goes OFF again.
Operand Data Areas:

St:
Starting Word

IO, AR, HR, LR

E:
End Word

IO, AR,
SHIFT REGISTER CONT...

Adress = 200.xx

<table>
<thead>
<tr>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Push Start PB

Select Auto Mode Start set I='1'

Auto/Manual

Start P

Stop R

ST: Start Word

200

E: End word

200

Select Auto Mode Start set I='1'
Basic Shift Register: exercise 8

- Start
- Select Auto Mode
- Push Start Button x2
- Green Light ON
- Push Start Button
- Yellow Light ON
- Push Start Button
- Red Light ON
- Push Stop Button
- All indicator Off
- End

©nikmzakmal 2010
Basic Shift Register cont...: exercise 9

[Diagram of a basic shift register with labeled elements and connections showing the flow of data through shifts 1 to 5, 2 to 7, and 3 to 12, with corresponding labels such as 'Shift Register', 'Start', and 'End word'.]
Interlocking (IL and ILC)

- Permissive to run a process

IL(002)

ILC(003)
Interlocking (IL and ILC)