Traction control system

(anti-skid)

1. Periferals

To ensure the solution of the problem of developing an anti-skid microprocessor system (MPS) for a car, we will select the necessary component base. In the designed microprocessor system, we will use the following components:

1. MPU (Microprocessor Unit) - for executing the control program for the operation of all devices within the MPS.
2. Memory - containing the executable program as well as all the data required for the operation of the MPU.
3. Timer - for generating a 0.5-second time interval and counting pulses from the car's wheel speed sensors.
4. Programmable Interrupt Controller - for "tracking" the moment when the 0.5-second time interval ends.
5. SPI bus- for issuing a digital code to control the feedback pump, switching valve, high-pressure valve, and for reading measurements from the brake force sensor.

2. Structural description

In a simplified manner, the operation of the microprocessor system (MPS) is as follows:

After applying power, the Timer generates an impulse at a 0.5-second interval to trigger an interrupt subroutine.

Within the interrupt subroutine, the MPS reads the number of pulses received from the speed sensors of both the driving and non-driving wheels. Assuming the wheel's diameter is D=60cm, one wheel revolution corresponds to a length of L = πD = 3.14 * 0.6 = 1.9m. At a speed of V=80km/h=22m/s, the car's wheels will make 22/1.9 = 12 revolutions per second. Therefore, at this speed, the Timer will receive 120 pulses every second, or 60 pulses every 0.5 seconds. The MPS then estimates the number of pulses (N) received by the Timer within 0.5 seconds.

If N > 60, no actions are taken. Otherwise, the MPS detects wheel slip based on the difference in measured wheel speeds.

In the case of wheel slip, the MPS issues a code to activate the feedback pump, close the Programmable Interrupt Controller, open the SPI bus, and take further actions.

Subsequently, the MPS reads the value of braking force from the brake force sensor. If the value reaches 80, the feedback pump is turned off. Afterward, with a 0.5-second interval, the MPS measures the speeds of the driving and non-driving wheels of the car. When their values are aligned, the MPS issues a digital code to open the Programmable Interrupt Controller and close the high-pressure valve. The MPS then repeats this cycle of actions.

3. Functional scheme and description

The functional diagram of the anti-skid microprocessor system is based on a previously developed structural diagram and is presented in Figure 1.

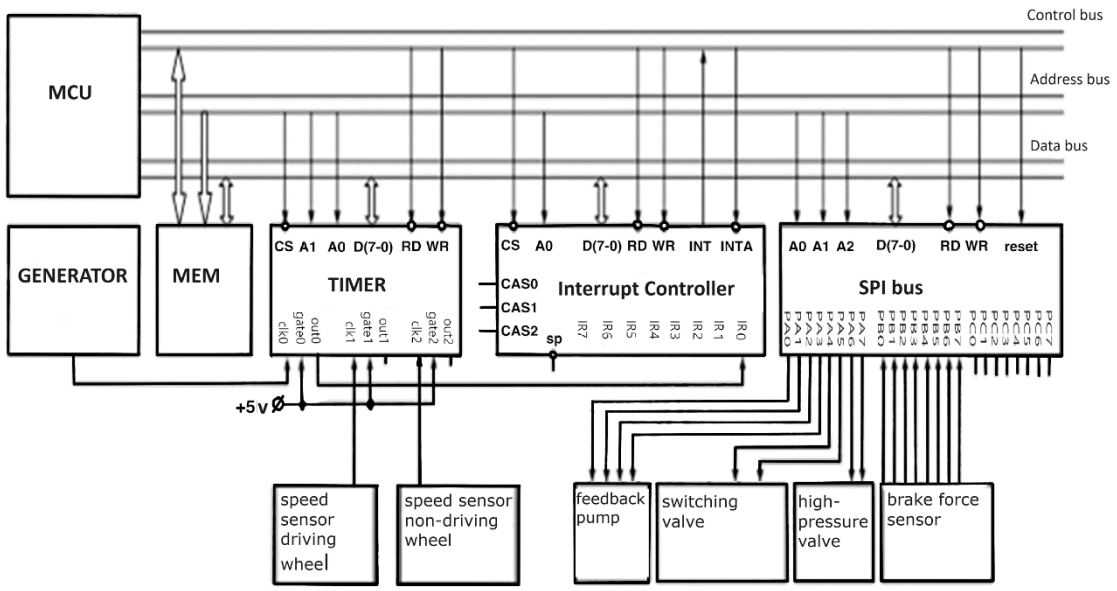

Figure 1. Microprocessor Anti-Skid System for an Automobile.

Functional electrical diagram.

The operation of the diagram is described as follows: After applying power, a signal from a 10kHz pulse generator is sent to clk0 of the Timer, where 0.5-second duration pulses are generated. These pulses, when applied to Input Register 0 of the Programmable Interrupt Controller, trigger interrupt requests. Simultaneously, pulses from the speed sensors of the driving and non-driving wheels are sent to Counters 1 and 2 of the Timer. If the wheel rotation speed is above 12 revolutions per second (equivalent to 80 km/h), the Microprocessor System (MPS) terminates the interrupt subroutine.

In the case where the car's speed is within the range of 0 to 80 km/h, the MPS calculates the difference in the rotation speeds of the driving and non-driving wheels. Based on this difference (for simplification, we consider wheel slip to occur when the speeds of the driving and non-driving wheels differ by 1), the presence of wheel slip is determined. Upon detecting wheel slip, the MPS sends a code to Port A of the SPI bus, which triggers the activation of the feedback pump, the closing of the switching valve, and the opening of the high-pressure valve.

Next, the MPS continuously reads the content of Port B of the SPI bus, to which the brake force sensor is connected. If the brake force exceeds 80, a code is sent to Port A, resulting in the deactivation of the feedback pump. Subsequently, the MPS waits for the alignment of the speeds of the driving and non-driving wheels. Upon alignment, the MPS sends a code to Port A of the SPI bus, leading to the opening of the switching valve and the closing of the high-pressure valve. The MPS then repeats this cycle of actions.

4. Control words

To ensure the functionality of the components within the developed system, control words will be generated for initializing the Programmable Interrupt Controller and the SPI bus.

To form the control word for Programmable Interrupt Controller, the address of the start of the memory area where the interrupt subroutine is located needs to be known. Let's assume this address is 3000(16). The control word for initializing Programmable Interrupt Controller will be as follows:

Control Word 1 (СКИ1) = 00010110(2) = 16(16)

Control Word 2 (СКИ2) = 00110000(2) = 30(16)

Loading these control words results in Programmable Interrupt Controller automatically providing the microprocessor with the address of the interrupt subroutine when a logical '1' signal appears at Input Register 0 (IR0), and the microprocessor takes control.

Now, let's create control words for initializing the operation of SPI bus. Since Port A operates as an output and Ports B and C function as inputs, the control word will be as follows:

SPI Control Word (УСППИ) = 10001011(2) = 8B(16).

Timers clk0 in the Programmable Interrupt Controller operates in mode "2," which means it divides the frequency by a constant N loaded into it. To generate a periodic signal with a period of 0.5 seconds, you need to load a constant N0=10000/2=1388(16) into it.

Timers 1 and 2 (clk1 and clk2) in the Programmable Interrupt Controller are used for counting pulses from the speed sensors of the driving and non-driving wheels (mode "0"). Therefore, you need to pre-load a constant N=255(10)=FF(16) into them.

Thus, the control words for the Programmable Interrupt Controller are as follows:

Control Word for clk0 = 00110100(2) = 34(16)

Control Word for clk1 = 01010000(2) = 50(16)

Control Word for clk2 = 10010000(2) = 90(16)

5. Algorithm

The algorithm is described as follows: Upon applying power to the Microprocessor, it awaits interrupt requests, which occur periodically at 0.5-second intervals. In the interrupt handling subroutine, the Microprocessor reads the content of clk2 in the Programmable Interrupt Controller 2 and calculates the number of pulses received from the speed sensor of the non-driving wheel. It then compares this value to the threshold characteristic of the car's speed, which is 80 km/h.

If the car's speed is less than 80 km/h, the Microprocessor reads the content of clk1 in the Programmable Interrupt Controller, calculates the number of pulses from the speed sensor of the driving wheel, and proceeds to check for wheel slip.

If wheel slip is detected, the Microprocessor activates the feedback pump, closes the switching valve, and opens the high-pressure valve. Subsequently, the Microprocessor cyclically queries the brake force sensor until its value exceeds 80. Upon reaching this specified value, the feedback pump is deactivated, and then the cyclically measures the number of pulses (N1, N2) received from the speed sensors of the driving and non-driving wheels, which enter clk1 and clk2 of the Programmable Interrupt Controller over a 0.5-second interval.

After comparing N1 and N2, if N1 = N2, the cyclically opens the switching valve and closes the high-pressure valve. Subsequently, the interrupt subroutine is concluded.

6. Assembly Program Listing

Initialization Subroutine

```
MVI A, 30          ; Load constant 30(16) into the accumulator
OUT C1             ; Load into control word register СКИ2
MVI A, 16          ; Load constant 16(16) into the accumulator
OUT C0             ; Load into control word register СКИ1
MVI A, 8B          ; Load constant 8B(16) into the accumulator
OUT FF             ; Load into SPI control word register ППИ
MVI A, 34          ; Load constant 34(16) into the accumulator
OUT EF             ; Load into control word for Timer 0 БИС ПИТ
MVI A, 88          ; Load constant 1388(16) into the accumulator
OUT EC             ; Load constant into CT0 of БИС ПИТ
MVI A, 13          ; Load constant 1388(16) into the accumulator
OUT EC             ; Load constant into CT0 of БИС ПИТ
MVI A, 50          ; Load constant 50(16) into the accumulator
OUT EF             ; Load control word for Timer 1 БИС ПИТ
MVI A, FF          ; Load constant FF(16) into the accumulator
OUT ED             ; Load constant into CT1 of БИС ПИТ
```

```
MVI A, 90          ; Load constant 90(16) into the accumulator
OUT EF             ; Load control word for Timer 2 БИС ПИТ
MVI A, FF          ; Load constant FF(16) into the accumulator
OUT EE             ; Load constant into CT2 of БИС ПИТ
MOV D, 0           ; Set the mode for the first wheel rotation speed reading (D=1
– waiting for wheel slip to end)
```

Interrupt Subroutine IR0

```
IN EE              ; Input the content of CT2 into the accumulator
MOV B, A           ; (B) = (A)
MVI A, FF          ; (A) = FF
SUB B              ; (A) = (A) – (B)
CPI 3C             ; Compare the accumulator with 3C(16) = 60(10)
JP EXIT            ; If (A) > 3C(16), jump to the label EXIT
MOV B, A           ; (B) = (A)
IN ED              ; Input the content of CT1 into the accumulator
MOV C, A           ; (C) = (A)
M0: MVI A, FF   ; (A) = FF
SUB C              ; (A) = (A) – (C)
MOV C, A           ; (C) = (A)
MOV A, D           ; (A) = (D)
CPI 00             ; Compare the accumulator with 00
JNZ M1             ; If the accumulator is not equal to 0, jump to M1
MOV A, C           ; (A) = (C)
SUB B              ; (A) = (A) – (B)
CPI 01             ; Compare the accumulator with 01
JM EXIT            ; If (A) - (B) < 1, jump to the EXIT label
MVI A, 31          ; (A) = 31(16). Code to activate the feedback pump, close the
switching valve, and open the high-pressure valve
```

```
OUT FC          ; Output the contents of the accumulator to Port A
WR: IN FD       ; Read the content of Port B into the accumulator
CPI 50          ; Compare the accumulator with 50(16) = 80(10)
JM WR           ; If (A) < 80(10), jump to the label WR
MVI A, C1       ; (A) = C1. Code to deactivate the feedback pump
OUT FC          ; Output the contents of the accumulator to Port A
MVI D, 01       ; (D) = 01
JMP EXIT        ; Unconditional jump to the label EXIT
M1: MOV A, C    ; (A) = (C)
CMP B           ; Compare the accumulator with register B
JNZ EXIT        ; If they are not equal, jump to the label EXIT
MVI A, C4       ; If they are equal, (A) = C4. Code to open the switching valve
and close the high-pressure valve
OUT FC          ; Output the contents of the accumulator to Port A
MVI D, 00       ; (D) = 0
EXIT: RET       ; Return from the interrupt
```