COP472-3

AN-749 Quadrature Signal Interface to a COP400 Microcontroller

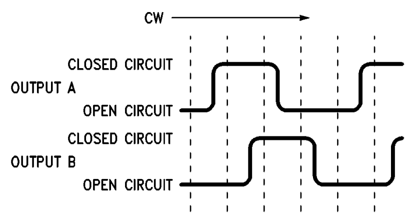
Literature Number: SNOA211

Quadrature Signal Interface to a COP400 Microcontroller

National Semiconductor
Application Note 749
Walter Bacharowski
February 1991

INTRODUCTION

Switches have always been a popular way of getting information into a microcontroller. Two-bit quadrature output devices, also known as two-bit gray code output, use two switches that are mechanically coupled together thru a shaft so that as the shaft is rotated the switches generate two square waves that are 90 degrees out of phase with each other. This is also known as being in quadrature, see *Figure 1*. The reason for doing this is that within the two signals there is the information to detect the direction of rotation, i.e., clockwise (CW) or counterclockwise (CCW). This type of device allows an input variable to be increased or decreased by CW or CCW rotation of the shaft. Additionally, these devices allow continuous rotation in either direction, which lets the span and resolution of the input variable to be a function of the software.



TL/DD/11147–1

FIGURE 1

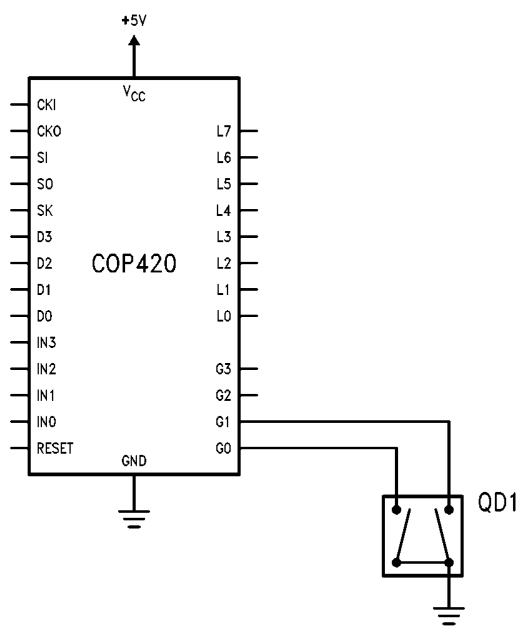
OPERATION

Figure 2 shows a hardware connection of a quadrature output device to the COP400 microcontroller. Although in this example the G0 and G1 I/O pins are used, any pin that can be used as an input could be used with the appropriate changes in the software.

In this example the output of device QD1 is processed to detect a state change in the quadrature signal and which direction the change was in. A 3-digit BCD variable, which is stored in RAM, is then incremented or decremented. The variable is defined to have a range of 200 to 350 units. The routine allows the variable to saturate at it's upper and lower limits when reached.

Figure 3 displays the two waveforms that are generated by QD1 as it's shaft is rotated from an arbitrary starting position. Each edge represents a change of state. By keeping track of the state that was moved from and the state that currently exists, it can be determined which direction the rotation was in.

Referring to *Figure 3*, there are 4 possible states for a starting position, (00, 01, 11, 10), and they will be referred to as the previous state. There are also 4 possible states to move to, (00, 01, 11, 10), and they will be referred to as the current state. *Figure 4* lists the 8 possible combinations of bits that can be formed by starting from each previous state and rotating CW or CCW to the current state. If the two bits of the previous state and current state are concatenated into one 4-bit value, each value will be unique. The routine



TL/DD/11147–2

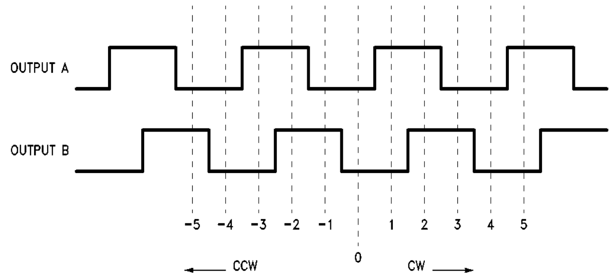
FIGURE 2

 RRD-B30M75/Printed in U. S. A.

POSITION	OUTPUT STATE A	B
−5	0	1
−4	0	0
−3	1	0
−2	1	1
−1	0	1
0	0	0
1	1	0
2	1	1
3	0	1
4	0	0
5	1	0

CCW CW

TL/DD/11147–4



TL/DD/11147–3

FIGURE 3. Quadrature Signal Output (Gray Code)

used in this example assigns the two bits of the current state to the low 2 bits of a 4-bit value, and the 2 bits of the previous state to the high 2 bits of a 4-bit value. This 4-bit value (see the column under the PS/CS heading in *Figure 4*) is then used as a pointer into a jump table which branches to the add or subtract part of the routine. This method takes advantage of the ''jump indirect'' instruction which implements a multiway branch based on the value of a pointer.

The routine to input data from the quadrature device reads the value of G0 and G1 and compares it to the value stored from the previous read operation. If the two values are equal there is no input to process. If the two values are not equal there is an input and the data is processed to determine if one is to be added to or subtracted from the variable.

The flow chart details the operation of the subroutine ''QUAD''.

In *Figure 4*, only 8 of the possible 16 combinations are used. To account for potential spurious operation if one of the 8 undefined combinations occur, they are ignored by this routine by branching to a return instruction which bypasses any additional processing.

The source listing for this example subroutine, which is named ''QUAD'', is provided. An initialization routine that is required to set up the starting parameters is also included.

CONCLUSION

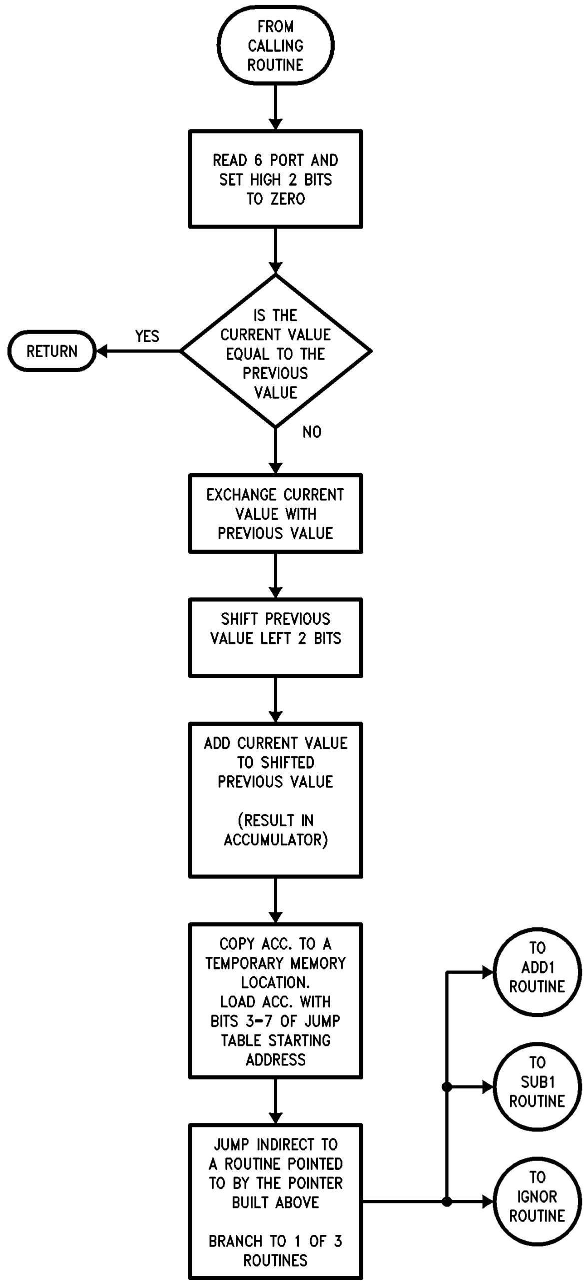
This application note demonstrates the relative ease of interfacing a quadrature device to a COP400 microcontroller. The combination of a low cost microcontroller and input device can provide the basis for a cost effective instrument or appliance design.

Direction	PS/CS	Hex Value	Operation
CW	00 10	2	Add 1
CCW	00 01	1	Subtract 1
CW	01 00	4	Add 1
CCW	01 11	7	Subtract 1
CW	11 01	D	Add 1
CCW	11 10	E	Subtract 1
CW	10 11	B	Add 1
CCW	10 00	8	Subtract 1

PS = Previous State
CS = Current State

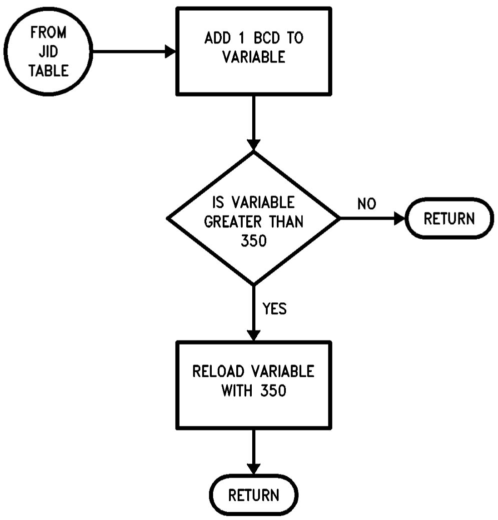
FIGURE 4

QUAD



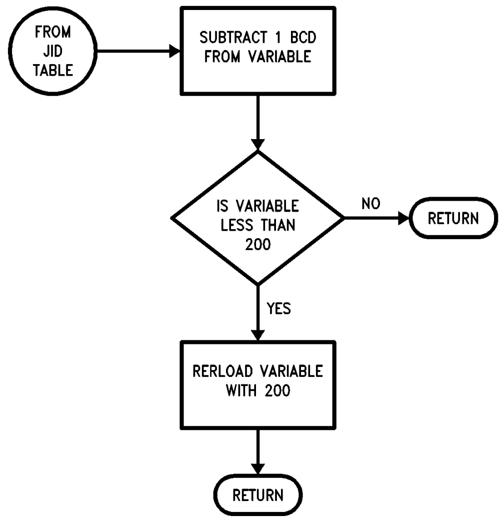
TL/DD/11147–5

ADD1



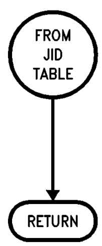
TL/DD/11147–6

SUB1



TL/DD/11147–7

IGNOR



TL/DD/11147–8

```
NATIONAL SEMICONDUCTOR CORPORATION
COP400 CROSS ASSEMBLER,REV:D,8 MAY 85
QUAD

                  ;
                  ;                     QUAD.MAC
                  ;                     QUADRATURE SIGNAL INTERFACE TO THE COP420
                  ;                     AUGUST 1, 1990
                  ;                     WALTER BACHAROWSKI
                  ;
                                .TITLE QUAD
                                .CHIP 420
                          ;
                          ;********
                          ;
                          ;ASSIGNMENTS
                          ;
                          ;********
                          ;
       000F               PV     =      0,15    ;PREVIOUS VALUE REGISTER
       003D               SCR0   =      3,13    ;SCRATCH PAD LOCATION
       003E               SCR1   =      3,14    ;SCRATCH PAD LOCATION
       003F               SCR2   =      3,15    ;SCRATCH PAD LOCATION
       001D               VLD    =      1,13    ;VARIABLE VALUE, LOW DIGIT
       001E               VMD    =      1,14    ;VARIABLE VALUE, MIDDLE DIGIT
       001F               VHD    =      1,15    ;VARIABLE VALUE, HIGH DIGIT
                          ;
                          ;***********
                          ;
                          ;PROGRAM START
                          ;
                          ;***********
                          ;
                  POR:
  0000 00                 CLRA
  0001 335F               OGI    15     ;SET G PORT TO 1's SO THEY CAN BE USED AS INPUTS.
  0003 0E                 LBI    PV     ;GET INITIAL SETTING OF QUADRATURE DEVICE
  0004 332A               ING           ;AND STORE IN PREVIOUS VALUE REGISTER
  0006 06                 X
  0007 42                 RMB    2      ;MASK HIGH 2 BITS
  0008 43                 RMB    3
                          ;
  0009 1C                 LBI    VLD    ;LOAD MINIMIUM VALUE FOR THE VARIABLE
  000A 70                 STII   0
  000B 70                 STII   0
  000C 72                 STII   2
                          ;
                  IDLE:
  000D 6A11               JSR    QUAD   ;CONTINUOUS LOOP TO CHECK THE INPUT
  000F CD                 JP     IDLE
                          ;
                          ;
       0200               . =    X'200  ;FORCE THE QUAD ROUTINE TO START AT HEX 200
                          ;
                          ;
```

TL/DD/11147–9

```
NATIONAL SEMICONDUCTOR CORPORATION
COP400 CROSS ASSEMBLER,REV:D,8 MAY 85
QUAD

                           ;******
                           ;
                           ;START OF JUMP TABLE FOR PROCESSING INPUTS
                           ;
                           ;******
                           ;
                  QUADJT:                     ;POINTER VALUE IN HEX
    0200 10                .ADDR   IGNOR   ;0
    0201 28                .ADDR   ADD1    ;1
    0202 3D                .ADDR   SUB1    ;2
    0203 10                .ADDR   IGNOR   ;3
    0204 3D                .ADDR   SUB1    ;4
    0205 10                .ADDR   IGNOR   ;5
    0206 10                .ADDR   IGNOR   ;6
    0207 28                .ADDR   ADD1    ;7
    0208 28                .ADDR   ADD1    ;8
    0209 10                .ADDR   IGNOR   ;9
    020A 10                .ADDR   IGNOR   ;A
    020B 3D                .ADDR   SUB1    ;B
    020C 10                .ADDR   IGNOR   ;C
    020D 3D                .ADDR   SUB1    ;D
    020E 28                .ADDR   ADD1    ;E
    020F 10                .ADDR   IGNOR   ;F     END OF JUMP TABLE
                           ;
                  IGNOR:                          ;BYPASS ANY ADDITIONAL PROCESSING
    0210 48                RET
                           ;
                           ;********
                           ;
                           ;PROCESS INPUT TO CHECK FOR A CHANGE OF STATE
                           ;
                           ;********
                           ;
                  QUAD:
    0211 3E                LBI     SCR2            ;GET CURRENT INPUT STATE
    0212 332A              ING                     ;AND MASK HIGH TWO BITS
    0214 06                X                       ;THEN COMPARE PREVOIUS AND CURRENT
    0215 42                RMB     2               ;STATE
    0216 43                RMB     3               ;
    0217 35                LD      3               ;COPY MASKED VALUE TO ACCUM. AND POINT
    0218 21                SKE                     ;TO PREVIOUS STATE. CHECK IF EQUAL
    0219 DB                JP QUAD2
    021A 48                RET                     ;THEY ARE EQUAL SO RETURN
                           ;
                  QUAD2:
    021B 36                X       3               ;EXCHANGE CURRENT AND PREVIOUS VALUES
    021C 06                X                       ;AND POINT TO SCRATCH LOCATION
    021D 00                CLRA
    021E 31                ADD                     ;DO A LEFT SHIFT OF 2 BITS
    021F 31                ADD                     ;THIS FORMS THE 2 HIGH BITS OF THE
    0220 31                ADD                     ;JUMP POINTER
```

TL/DD/11147–10

```
0221 31              ADD
                     ;
0222 0E              LBI     PV               ;NOW ADD THE CURRENT STATE THAT WAS JUST
0223 31              ADD                      ;PROCESSED. BECOMES THE 2 LOW BITS OF
                                              ;THE JUMP POINTER
0224 3E              LBI     SCR2             ;SET UP THE POINTER FOR THE
0225 06              X                        ;JUMP INDIRECT POINTER
0226 00              CLRA
                     ;
0227 FF              JID                      ;BRANCH TO REQUIRED ROUTINE
                     ;
                     ;********
                     ;
                     ;ADD 1 TO THE VALUE OF THE VARIABLE AND CHECK FOR ITS MAX
                     ;VALUE.
                     ;
                     ;********
                     ;
             ADD1:
0228 1C              LBI     VLD              ;POINT TO LEAST SIGNIFICANT DIGIT
0229 22              SC                       ;USE CARRY TO ADD 1
             ADD1L:
022A 00              CLRA
022B 56              AISC    6                ;BCD CORRECTION
022C 30              ASC
022D 4A              ADT
022E 04              XIS                      ;STORE DIGIT AND POINT TO NEXT DIGIT
022F EA              JP      ADD1L
0230 3C              LBI     SCR0             ;STORE VALUE TO CHECK FOR MAX VALUE
0231 79              STII    9
0232 74              STII    4
0233 76              STII    6
0234 6A53            JSR     ADDLIM
0236 20              SKC                      ;IF CARRY IS SET ON RETURN FROM
0237 48              RET                      ;ADDLIM THEN THE VARIABLE IS LARGER
0238 1C              LBI     VLD              ;THEN ITS MAXIMIUM VALUE
0239 70              STII    0                ;SO RESET IT TO ITS MAX VALUE
023A 75              STII    5
023B 73              STII    3
023C 48              RET
                     ;
                     ;********
                     ;
                     ;SUBTRACT ONE FROM THE VARIABLE AND CHECK FOR
                     ;IT BEING GREATER THEN THE MINIMIUM VALUE.
                     ;
                     ;********
                     ;
             SUB1:
023D 1C              LBI     VLD     ;SUBTRACT 1 BY FORCING A BORROW
023E 32              RC
```

TL/DD/11147–11

```
NATIONAL SEMICONDUCTOR CORPORATION
COP400 CROSS ASSEMBLER,REV:D,8 MAY 85
QUAD

                  SUB1L:
     023F 00              CLRA


     0240 10              CASC
     0241 4A              ADT             ;BCD CORRECTION
     0242 04              XIS             ;STORE DIGIT AND POINT TO NEXT DIGIT
     0243 623F            JMP     SUB1L
                          ;
     0245 3C              LBI     SCR0    ;STORE VALUE TO CHECK LOWER LIMIT OF THE VARIABLE
     0246 70              STII    0
     0247 70              STII    0
     0248 78              STII    8
     0249 6A53            JSR     ADDLIM
                          ;
     024B 20              SKC             ;IF CARRY IS NOT SET ON RETURN THEN VARIABLE
     024C CE              JP      SUB2    ;LESS THEN IT'S MINIMIUM VALUE
     024D 48              RET
                  SUB2:
     024E 1C              LBI     VLD     ;FORCE VARIABLE TO ITS MIN VALUE
     024F 70              STII    0
     0250 70              STII    0
     0251 72              STII    2
     0252 48              RET
                          ;
                          ;********
                          ;
                          ;ADD A VALUE STORED IN SCR0 TO SCR3 TO THE VALUE OF THE
                          ;VARIABLE NONDESTRUCTIVELY. THE STATE OF THE CARRY BIT
                          ;IS USED BY THE CALLING ROUTINE AS A RESULT.
                          ;
                          ;********
                          ;
                  ADDLIM:
     0253 1C              LBI     VLD
     0254 32              RC
                  ADLIM1:
     0255 25              LD      2       ;BCD ADDITION OF 3 DIGITS
     0256 56              AISC    6
     0257 30              ASC
     0258 4A              ADT
     0259 24              XIS     2       ;PROCESS NEXT DIGIT
     025A D5              JP      ADLIM1
     025B 48              RET
                          ;
                          .END
```

TL/DD/11147–12

```
NATIONAL SEMICONDUCTOR CORPORATION          PAGE:     5
COP400 CROSS ASSEMBLER,REV:D,8 MAY 85
QUAD
SYMBOL TABLE


ADD1    0228     ADD1L   022A     ADDLIM  0253     ADLIM1  0255
IDLE    000D     IGNOR   0210     POR     0000 *   PV      000F
QUAD    0211     QUAD2   021B     QUADJT  0200 *   SCR0    003D
SCR1    003E *   SCR2    003F     SUB1    023D     SUB1L   023F
SUB2    024E     VHD     001F *   VLD     001D     VMD     001E *

NO ERROR LINES

  108  ROM BYTES USED

COP  420  ASSEMBLY

SOURCE CHECKSUM = 2177
OBJECT CHECKSUM = 01FF

INPUT   FILE C:QUAD.MAC
LISTING FILE C:QUAD.PRN
OBJECT  FILE C:QUAD.LM
```

TL/DD/11147–13

National Semiconductor Corporation
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: 1(800) 272-9959
TWX: (910) 339-9240

National Semiconductor GmbH
Livry-Gargan-Str. 10
D-82256 Fürstenfeldbruck
Germany
Tel: (81-41) 35-0
Telex: 527649
Fax: (81-41) 35-1

National Semiconductor Japan Ltd.
Sumitomo Chemical Engineering Center
Bldg. 7F
1-7-1, Nakase, Mihama-Ku
Chiba-City,
Ciba Prefecture 261
Tel: (043) 299-2300
Fax: (043) 299-2500

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductores Do Brazil Ltda.
Rue Deputado Lacorda Franco
120-3A
Sao Paulo-SP
Brazil 05418-000
Tel: (55-11) 212-5066
Telex: 391-1131931 NSBR BR
Fax: (55-11) 212-1181

National Semiconductor (Australia) Pty, Ltd.
Building 16
Business Park Drive
Monash Business Park
Nottinghill, Melbourne
Victoria 3168 Australia
Tel: (3) 558-9999
Fax: (3) 558-9998