# 6 M68HC11 INTERNAL PERIPHERAL INTERFACE

## 6.1 Internal Peripheral Interface in General

- **Peripheral device / input-output** – tools to exchange data with the microcomputer. Example: switch, LED, printer, keyboard, monitor, disk drives and etc. Because the different characteristics & speed between peripheral device & CPU, peripheral device is not connected directly to microcomputer/microcontroller. Requires medium between microcontroller & peripheral.

- M68HC11 have some type of interface in the chips to facilitate the microcontroller connection with external peripheral in the embedded systems.

  **Interface / parallel port** transfer some bit (usually 8) simultaneous data.
  **Series interface** transfer data in series bit by bit in the low speed devices as modem.
  **Analog to digital converter** is useful for interface with analog devices.
  **Timer** is useful for the precise timing, production of periodic signals & measurement frequency periodic signal.

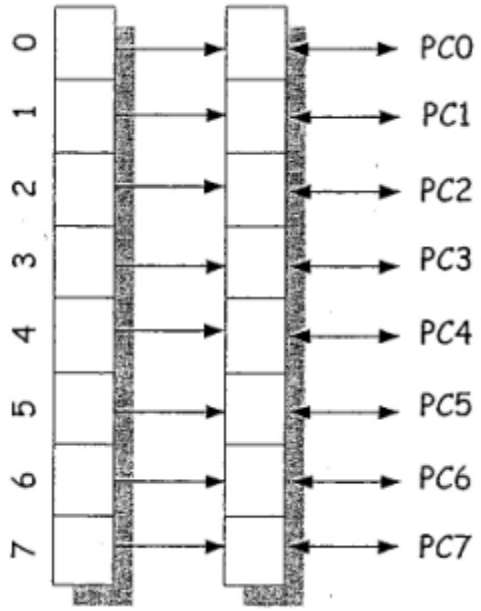
- The part of the 68HC11 internal interface system can be controlled through some registers that has been plotted. The register addresses located on the map address M68HC11 at address $1000-$103F. Mapping technique of this input-output system is called Memory Mapped Input-Output. So, any instruction and addressing modes that the processing of memory can be used to process the M68HC11 internal interface system. The use of various indexed recommended to process the registers!

## 6.2 Parallel Input – Output Medium

- I/O two-way port (C & D Port)
  Each two-way port has 2 registers that are related:
  1) Direction Data Register (DDR) – used to state the data flow direction (input @ output) for each pin.
  2) Data Register – register that store data read from pin or be written out to the pin.

Each pin at the port C & D is connected to certain bits in the DDR & data register. To make a pin/bit at the port C & D as output, the value 1 must be written in the DDR at the bit position. When reset the DDR content is 0, the port C & D used as input.

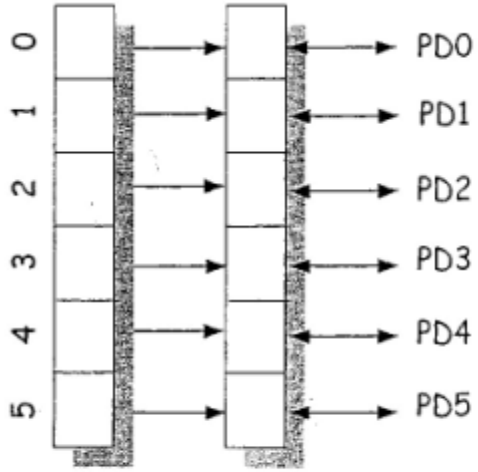
**Port C** – Single chip mode




Direction Data Register (DDRC) ($1007)

Data Register C ($1003)

**Port D** – All modes




Direction Data Register (DDRC) ($1009)

Data Register D ($100A)

- I/O two-way port (cont..)

Example: Write a program to read logic state at the pin PD0-PD4 and send the logic state to pin PC0-PC4.

Solution: must made bit 0-4 port D as input and bit 0-4 port C as output. Other bits do not care.

1) Use index modes to addressing register m/s

```
DFTRIO    EQU     $1000       ;alamat asas daftar m/k
PORTD     EQU     $08         ;ofset dftr data liang D
DDRD      EQU     $09         ;ofset dftr arah data D
PORTC     EQU     $03         ;ofset dftr data liang C
DDRC      EQU     $07         ;ofset dftr arah data C

          LDX     #DFTRIO
          LDAA    #%11100000  ;jadikan bit 0-4 sbg masukan
          STAA    DDRD,X      ;pd liang D
          LDAA    #%00011111  ;jadikan bit 0-4 sbg keluaran
          STAA    DDRC,X      ;pd liang D<
          LDAA    PORTD,X     ;baca dari PD0-PD4
          STAA    PORTC,X     ;hantar ke PC0-PC4
```

2) Use direct modes to addressing register m/s

```
PORTD     EQU     $1008       ;alamat dftr data liang D
DDRD      EQU     $1009       ;alamat dftr arah data D
PORTC     EQU     $1003       ;alamat dftr data liang C
DDRC      EQU     $1007       ;alamat dftr arah data C

          LDAA    #%11100000  ;jadikan bit 0-4 sbg masukan
          STAA    DDRD        ;pd liang D
          LDAA    #%00011111  ;jadikan bit 0-4 sbg keluaran
          STAA    DDRC        ;pd liang D
          LDAA    PORTD       ;baca dari PD0-PD4
          STAA    PORTC       ;hantar ke PC0-PC4
```

## 6.2 Parallel Input – Output Interface (cont…)

- **I/O PortA**

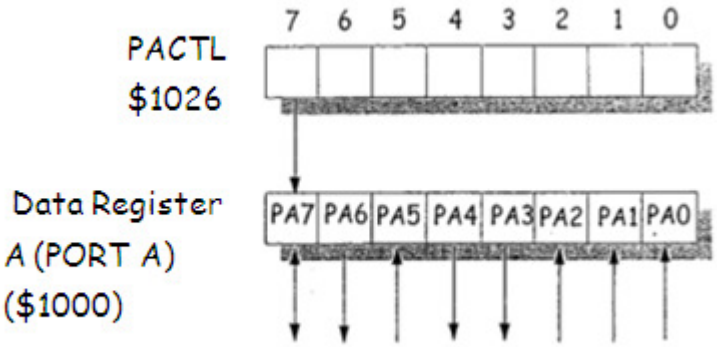
Bit PA0, PA1 & PA2 are only input pins. Write to the pins not have any impacts.

Bit PA3, PA4, PA5 & PA6 are only output pins. Read from the pins will provide logic to the latch pin.

Bit PA7 is the I/O two-way pin.

Data register for port A located at address $1000. The register store data that are read from pins PA0-PA2/PA7 or data that be written out to PA3-PA6/PA7.

Because of the bit PA0-PA6 has the set of data flow, direction data register (DDR) is not needed. DDR for bit PA7 is PACTL located at address $1026.

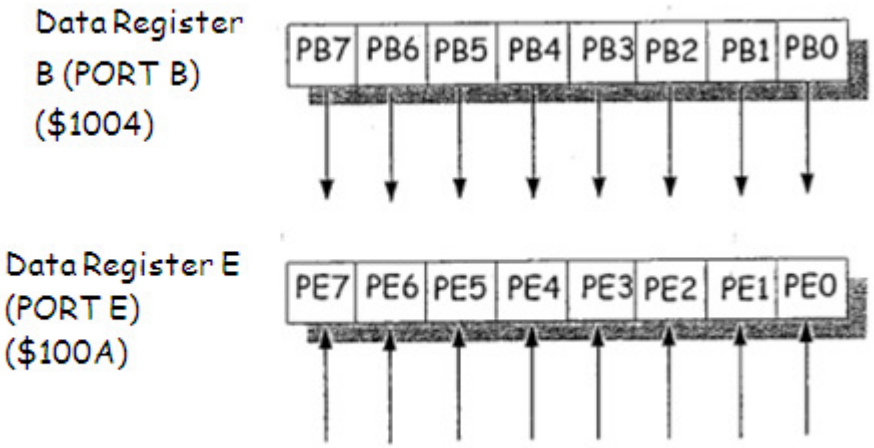



All of pins at port A associated with internal timer subsystem operation. The PA0-PA7 pins have alternative functions as input/ output to/from timer subsystem.

## 6.2 Parallel Input – Output Interface (cont..)

- Output port B & input port E
  Bit PB0-PB7 is only output pins. Read from the pins will provide logic at the latch pin. Output data register for port B located at address $1004.

  Bit PE0-PE7 is only input pins. Write to other pins not have any impact. Input data register for port E located at address $100A.

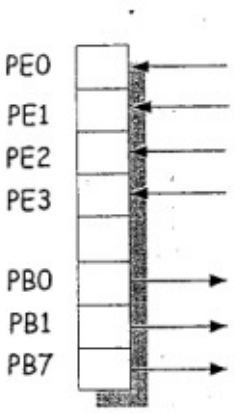



All of pins at port E is also input the analog to digital converter.

Example: The safety equipment based on M68HC11 has 4 intrusion detection sensors that are connected each to port E (PE0-PE4). 2 alarms are connected to port B (PB0-PB1) and one light is connected to PB7. The operation of the tool is: If 2 sensors at PE0-PE1 are activated, activate the alarm at PB0. If 2 sensors at PE2-PE3 activated, activate the alarm at PB1. If all of the sensors are activated, sound all of the alarms and activate the light. In normal case all of the alarms and lights are not activated. Assume all sensors, alarms and lights are high active logic devices.

## 6.2 Parallel Input – Output Interface (cont..)

**Solution:**



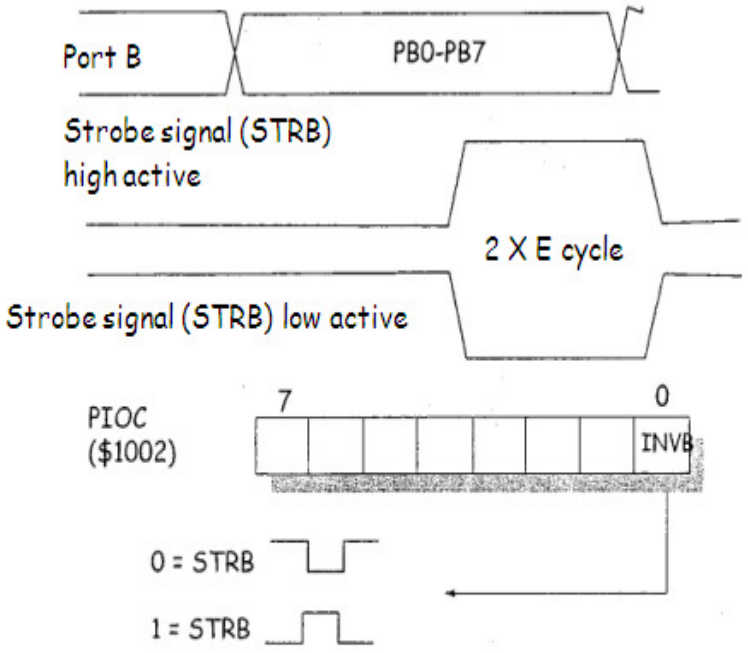

| Input | | | | Output | | |
|---|---|---|---|---|---|---|
| PE0 | PE1 | PE2 | PE3 | PB0 | PB1 | PB7 |
| | | | | | | |

## 6.2 Parallel Input – Output Interface (cont..)

- **Strobe B output port**

  Other than a simple output port, port B also can handle as strobe output port. In the strobe modes, pin STRB become a strobe output (one impulse that its width 2 cycles E) which is activated each time write operation to port B is doing.

  Usually the strobe signal is used by external peripheral to latch data that are sends from port B. The strobe signal can be program to be low active or high active through PIOC register ($1002) bit 0.

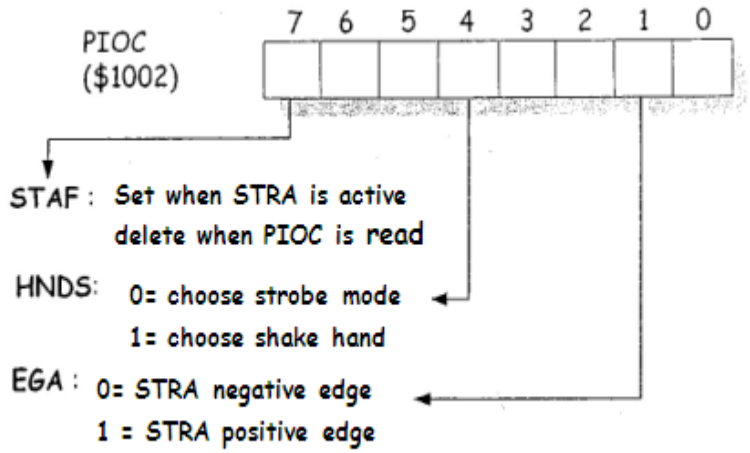



```
LDX     #$1000
BCLR    PIOC,X $01 ;jadikan STRB aktif rendah
LDAA    #$70
STAA    PORTB,X    ;hantar data terstrob ke PORTB
:
```

## 6.2 Parallel Input – Output Interface (cont..)

- **Strobe C input port**

  Other than become a simple input port, port C also can handle as strobe input port and shake hands input port. In the strobe modes, pin STRA become a trigger edge strobe input to 68HC11. STRA can be programmed as positive trigger edge or negative through PIOC register ($1002). When some edge was detected at STRA, logic state at port C (PC0-PC7) will be latch into register PORTCL & STAF flag in the PIOC register will be set. The 68HC11 can detect the presence of data through STAF flag and read the data that was latch through PORTCL.

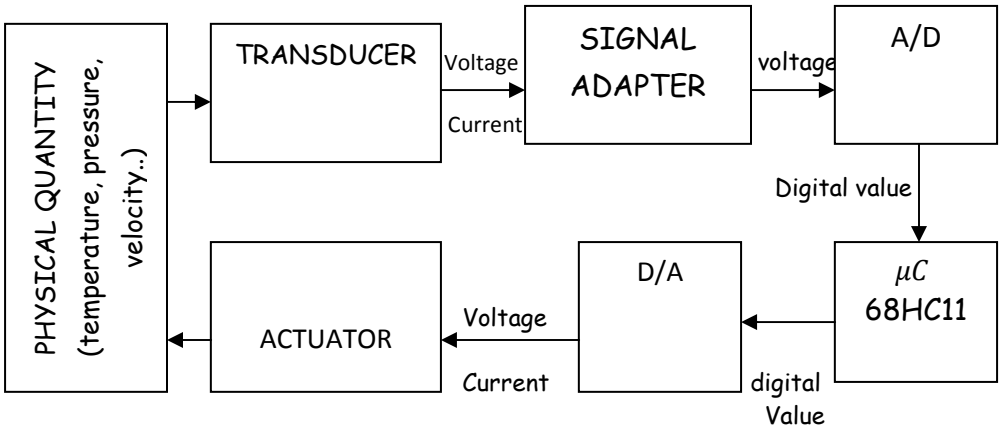



**Example**: read data from port C in the strobe mode and send data to the port B, continuously.

```
          LDX     #$1000
          LDAA    #0
          STAA    DDRC,X     ;jadikan PORTC masukan
          LDAA    PIOC,X     ;padam bit STAF jika diset
          BCLR    PIOC,X %00010010   ; HNDS=0,EGA=0
TUNGGU:   BRCLR   PIOC,X $80 TUNGGU  ;STAF=1?
          LDAA    PORTCL,X   ;baca dari liang C
          STAA    PORTB,X    ;hantar ke liang B
          JMP     TUNGGU     ;ulang
```

## 6.3 Analog Interface – Analog to Digital Converter (ADC)

- **Basic concepts of Analog – Digital – Analog Converter**



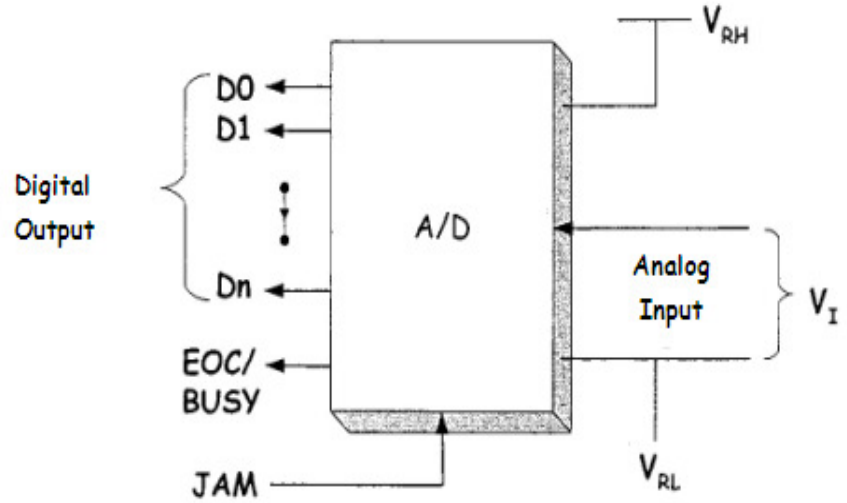

**A/D Converter** – convert analog signal to digital signal.

- In the M68HC11

**D/A Converter** – convert digital signal to analog signal.

- Not discussed

**A/D converter block diagram in general**



## 6.3 Analog Interface – A/D Converter (cont..)

- **Basic concept of Analog – Digital Converter (..)**

  **D0-Dn** is digital output n-bit that represents input analog. For the ADC chip standard:

| n | Name |
|---|---|
| 7 | ADC 8-bit ( 68HC11 internal ADC |
| 11 | ADC 12-bit |
| 15 | ADC 16-bit |

  **EOC/BUSY** is a signal that tells $\mu C$ whether ADC was ended turn analog signal to digital or is in the conversion process.

  Conversion time $(\tau_c)$ – the time required by ADC to convert one analog voltage to the digital code. $\tau_c$ have been determined by clock frequency that been used and the type of selected ADC. For converter 4 channels for 68HC11 $\tau_c$ is 128Xe (=64$\mu$sec with XTAL 8MHz).

$V_{RH}$ and $V_{RL}$are reference voltage (high & low) for ADC. The reference voltage must stable to produce accurate conversion.

$V_I$ is a voltage/input analog that want to convert to digital signal. The range maybe for $V_I$ is from $V_{RL}$ to $V_{RH}$. This range is called **Voltage Full-Scale** $V_{FS}$.

Resolution is a smallest change in the analog voltage to alter digital code of a bit.

$$\text{Resolution (R)} = \frac{V_{FS}}{2^{n+1}}$$

## 6.3 Analog Interface – A/D Converter (cont..)

- **Example:** calculate the resolution for 68HC11 internal ADC if $V_{RH}$ is connected to +5V and 0V. Compare with a single ADC 10-bit.

  **For 68HC11:**

  $V_{FS} = V_{RH} - V_{RL} = +5\text{V}$
  $Resolution(R) = \frac{5}{2^8} = 19.5\text{mV}$

  **For ADC 10-bit:**
  Resolution (R) = $\frac{5}{2^{10}} = 4.88\text{mV}$

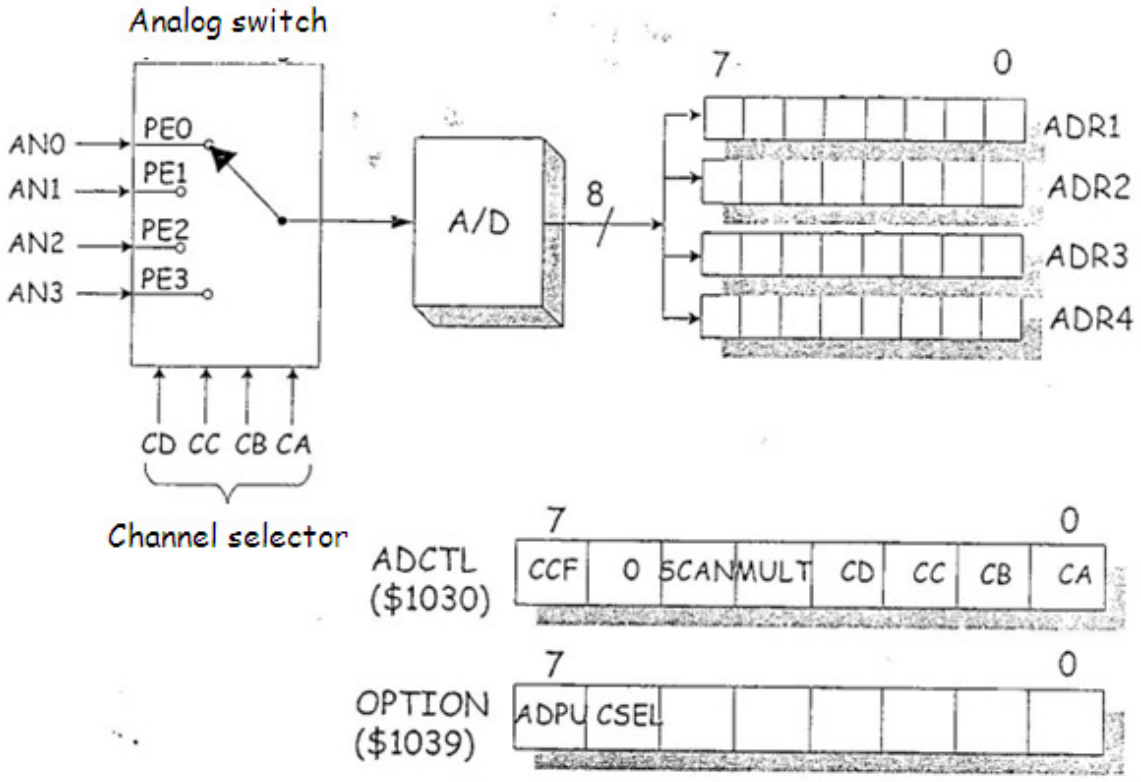
  ADC 10-bit have resolution that smaller than ADC 68HC11. It means that ADC 10-bit can measure the changes of voltage value as small as 4.88mV. The smallest changes of voltage that can be measure by ADC 68HC11 is 19.5mV. So that, the ADC 10-bit gives a more accurate reading.

  As compared with the ADC 68HC11, ADC 10-bit have the **highest** resolution.

  In general, the larger n, the resolution will be higher and more precise on converting analog to digital.

## 6.3 Analog Interface – A/D Converter (cont..)

- **Internal 68HC11 A/D Converter – Structure and Registers**




Registers that involved in the use of internal A/D:

ADR1, ADR2, ADR3, ADR4 ($1031, $1032, $1033, $1034) – 4 registers 8 bit that store digital code the conversion of A/D channel AN0-AN3. Digital code in the ADR1-ADR4 is 0 when analog voltage is equal to $V_{RL}$ and digital code is $FF when analog voltage equal to $V_{RH}$. Each additional or reduction of analog voltage as much as resolution will be change the digital code as much as a bit.

Exercise: For connection as in the example m/s 6/11, calculate the digital code that the result in the ADR1-ADR4 when analog voltage is:

a) +2.5V b) +4.0V

What is the value of analog voltage when the result of digital code is $50?

## 6.3 Analog Interface – A/D Converter (cont..)

- **Internal 68HC11 A/D Converter – Registers..**

ADCTL ($1030) – used to set the A/D operation mode, choose analog voltage channel to be changed and store A/D converter status.

Bit 0-3 (CA-CD) is used to choose analog channel to A/D converter.

| CD | CC | CB | CA | The selected of analog channel |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | AN0 |
| 0 | 0 | 0 | 1 | AN1 |
| 0 | 0 | 1 | 0 | AN2 |
| 0 | 0 | 1 | 1 | AN3 |

Bit 4 (MULT) select the A/D converter operation mode. Control the usage of one or multiple channels.
MULT = 0 – A/D converter operating in a single channel mode. In the single channel mode, analog voltage at the channel that is determined by bit 0-3 will be changed 4 times. The first conversion result in the ADR1, the second in the ADR2 and so on.
MULT = 1 – A/D converter operating in the multiple channels. In this mode, analog voltage in group 4 channels that is determined by bit 2-3 will be changed once. The selected of channel group and the destination of conversion result is given by the table following:

| CD | CC | CB | CA | Channel | Result register |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | PE0 | ADR1 |
| 0 | 0 | 0 | 1 | PE1 | ADR2 |
| 0 | 0 | 1 | 0 | PE2 | ADR3 |
| 0 | 0 | 1 | 1 | PE3 | ADR4 |
| | | | | | |
| 0 | 1 | 0 | 0 | PE4 | ADR1 |
| 0 | 1 | 0 | 1 | PE5 | ADR2 |
| 0 | 1 | 1 | 0 | PE6 | ADR3 |
| 0 | 1 | 1 | 1 | PE7 | ADR4 |

## 6.3 Analog Interface – A/D Converter (cont..)

- **Internal 68HC11 A/D Converter – Registers..**

  **ADCTL ($1030)..**

  **Bit 5 (SCAN)** selects the way of A/D converter.

  **SCAN = 0** - 4 A/D conversions are done once, after that conversion stop. The next conversion can be done with rewrite (STA) to ADCTL register.

  **SCAN = 1** -4 A/D conversions are done continuously. ADR1-ADR4 will be updated continuously with new data. The scanning mode.

  **Bit 6** – always 0

  **Bit 7 (CCF)** – conversion complete flag. CCF flag automatically set by $\mu C$ when conversion process of A/D completed. CCF automatically delete when ADCL register written. CCF = 1 show that 4 conversions are complete and data in the ADR1-ADR4 is valid. The programmer can test the CCF flag to know whether the conversion is complete or not.

  **OPTION ($1039)** – only bit 6 and 7 that have relation with the A/D converter.

  **Bit 7 (ADPU)** when set it to 1 will enable internal A/D converter. After the bit is set, the programmer must wait at least $100\mu sec$ to allow the circuit of internal A/D converter become stable.

  **Bit 6 (CSEL)** select clock source to internal A/D converter. When CSEL = 1, internal RC clock is selected and when CSEL = 0, E clock is selected as clock source.

## 6.3 Analog Interface – A/D Converter (cont..)

- **The steps to use the internal A/D converter:**

1) A/D converter enable and select clock source to A/D converter through OPTION register.
2) Wait at least 100$\mu sec$.
3) Select the wanted channel and A/D converter operation mode through ADCTL register. Write to ADCTL that will be start conversion.
4) Wait the conversion complete by test CCF flag. Read ADR1-ADR4 when the conversion is complete.

**Example**: 2 light intensity sensor detections is connected to PE0-PE1. The sensors show analog voltage (0-5V) to determine which sensor show the highest light intensity and store the value at address $50. Store the other one intensity at address $10. Assume both $V_{RH}$ and $V_{RL}$ is connected to +5V and 0V.

**Solution:**
2 analog sensors are located at channel 1 and 2. Select multiple channel modes (MULT = 1), non-scanning mode (SCAN = 0) and channel group PE0-PE3 in the ADCTL register. **ADCTL = %00010000**. The conversion result channel 1 in the ADR1 and channel 2 in the ADR2 .The highest intensity value can be determined with make comparison value in the ADR1 and ADR2.

Use index mode to achieve A/D converter registers.

## 6.3 Analog Interface – A/D Converter (cont..)

**Solution …**

*example program for SEM4233
*synopsis:
*determine the light intensity value at PE0 and PE1
*store largest value at $50 and other one value at $10
*programmer: DR. ROSBI MAMAT
*date: 31/08/2000

```
DFTRIO    EQU       $1000      ;alamat asas daftar m/k
ADCTL     EQU       $30        ;ofset hasil PE0
ADR1      EQU       $31        ;ofset hasil PE0
ADR2      EQU       $32        ;ofset hasil PE1
ADR3      EQU       $33        ;ofset hasil PE2
ADR4      EQU       $34        ;ofset hasil PE3
OPTION    EQU       $39        ;ofset dftr OPTION

          LDX       #DFTRIO
*pilih jam RC & bolehkan ADC dalaman
          BSET      OPTION,X  %11000000
          LDY       #30         ; tunggu 105 µs
LENGAH    DEY
          BNE       LENGAH

          LDAA      #%00010000  ;pilih rgm berbilang saluran
          STAA      ADCTL,X     ;& mulakan penukaran

TUNGGU    LDAA      ADCTL,X     ;uji bendera CCF
          BPL       TUNGGU      ;tunggu selagi tak selesai

          LDAA      ADDR1,X     ;simpan PE0 dlm A
          LDAB      ADDR2,X     ;simpan PE1 dlm B
          CBA                   ; (A) > (B) ?
          BHI       ABESAR      ; ya !
BBESAR    STAB      $50         ; simpan nilai terbesar
          STAA      $10
          BRA       TAMAT
ABESAR    STAA      $50         ; simpan nilai terbesar
          STAB      $10
TAMAT
          :
```

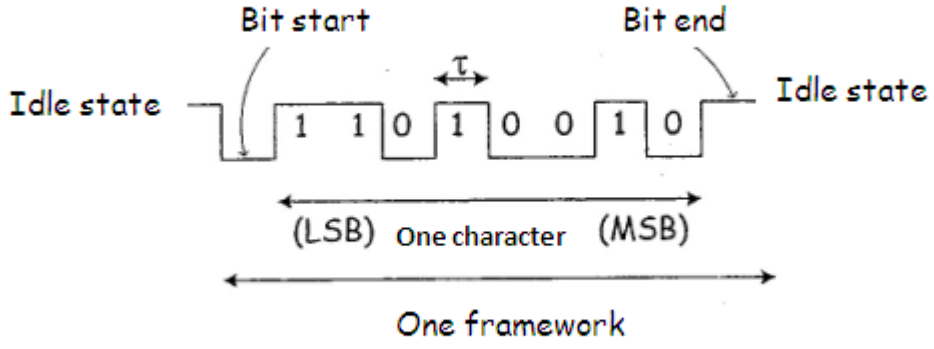
## 6.4 Serial Interface

- The serial data communication involved transmission/reception data bit by bit.
- Advantages of the serial communication: a small number of line/connection is needed than the parallel communication. 2 lines (send and receive) is needed in the full duplex asynchronous serial communication. 3 lines (send, receive, clock) is needed in the synchronous serial communication.
- Disadvantages of serial communication: need more time to send/receive data than the parallel communication.
- M68HC11 have 2 types of serial communication interface:

1) Serial Communication Interface (SCI)
2) Serial Peripheral Interface (SPI)

### 6.4.1 Serial Communication Interface (SCI)

- It is a asynchronous communication data
- Use 2 pins from port D:
  1) TXD/PD1 (pin 43) – as a line of serial data transmission
  2) RXD/PD0 (pin 42) – as a line of serial data reception
- Transmitted and received data in the serial framework below:




- SCI can send character with 8-bit or 9-bit. Usually MSB as parity bit.

### 6.4.1 Serial Communication Interface (SCI) (cont…)

- Each bit delivered within certain time, $\tau$, called bit time. The smaller bit time, the transmission data will be faster. The transmission data rate is called Baud Rate. Standard Baud rate are 300, 600, 1200, 2400, 4800, 9600, 19200

$$\text{Baud rate} = \frac{1}{\tau}$$

- In the 68Hc11, the Baud rate generated using clock E. the Baud rate at sender and receiver data must balance to avoid error.

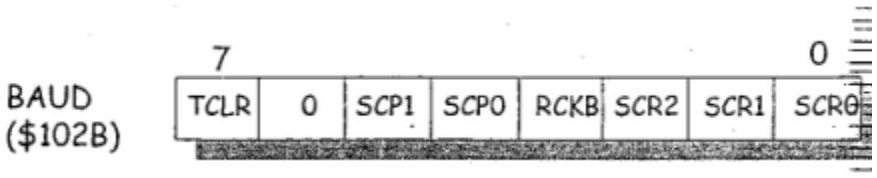
- The registers that involved in the SCI

**SCDR ($102F)** – register serial communication data. It is used to:

1. Store data 8-bit that to be sent out through SCI. Data is written into the SCDR will be shifted out in series through pin TXD.

2. Store data 8-bit received from SCI. Read from SCDR providing serial data that read from pin RXD.

**BAUD ($102B)** – register Baud rate. It is used to select Baud rate for SCI.




**Bit 3 & 7** (RCKB & TCLR) – it is used for Motorola factory test.

**Bit 4 & 5** (SCP0 & SCP1) – it is used as pre-scale (P) to divide clock E before become it as Baud rate generator. The value for pre-scale (P) is provided in the following table:

### 6.4.1 Serial Communication Interface (SCI) (cont…)

**BAUD ($102B)..**

| SCP1 | SCP0 | Clock E Divider (P) |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 3 |
| 1 | 0 | 4 |
| 1 | 1 | 13 |

**Bit 0, 1, 2** (SCR0-SCR2) – it is used to divide output from pre-scale before become it as Baud rate generator. The values for the divider (Q) are provided in the following table:

| SCR2 | SCR1 | SCR0 | Divider (Q) |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 4 |
| 0 | 1 | 1 | 8 |
| 1 | 0 | 0 | 16 |
| 1 | 0 | 1 | 32 |
| 1 | 1 | 0 | 64 |
| 1 | 1 | 1 | 128 |

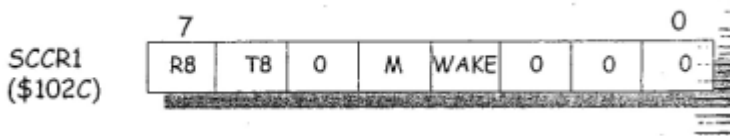
$$\text{Baud rate} = \frac{\text{Frek,Clock E}}{16\text{xPxQ}}$$

**Example:**
To get Baud rate 9600 with crystal 8MHz, select Q=1 and P=13. BAUD content=$30.

To get Baud rate 4800 with crystal 8MHz, select Q=2 and P=13. BAUD content=$31.

### 6.4.1 Serial Communication Interface (SCI) (cont…)

**SCCR1 ($102C)** – register SCI controller. It is used to set transmission/reception data framework format.



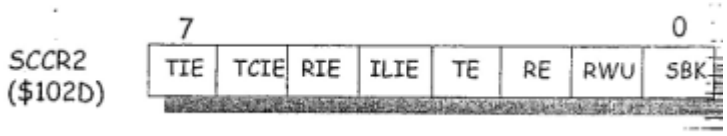

**Bit 6 & 7** (R8 & T8) - it is used when 9-bit data format is sent/ received. As an 8-bit storage, data that received (R8) or bit 8 data to be transmitted (T8).

**Bit 3** (WAKE) – only useful if 'wake-up' function is used.

**Bit 4** (M) – it is used to select data transmission/ reception format serial.
When **M=0**, data format=1 bit start, 8 bit data & 1 bit stop.
When **M=1**, data format=1 bit start, 9 bit data & 1 bit stop.

**Bit 0, 1, 2 & 5** – it is not used!

**SCCR2 ($102D)** – Register SCI controller. It is used to enable / disable transmission/ reception and interrupt that related to SCI.

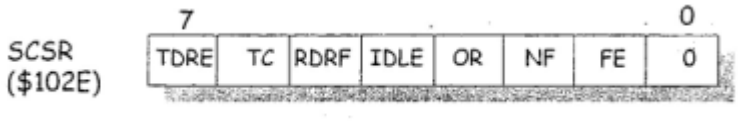



**Bit 0, 1, 4, 5, 6, 7** – it is used when interrupt facility and 'wake-up' function is needed. Not discussed here!

**Bit 2 & 3** – (RE & TE) – it is used to enable transmission (TE=1) and reception (RE=1) serial data. When RE=0 or TE=0, reception or transmission serial data disabled.

### 6.4.1 Serial Communication Interface (SCI) (cont…)

**SCSR ($102E)** – register SCI status. The bits in the SCSR provide data transmission/reception serial status and error occurs if there.




**Bit 7** (TDRE) – when set to 1 it show that data register for transmission is zero. It means that SCI ready to send further data.

**Bit 6** (TC) – when set to 1 it show that serial transmission data completed and TXD line achieved idle level.

**Bit 5** (RDRF) – when set to 1 it show that data register for reception is full. It means that SCI completed receive serial data from RXD line and new data located in SCDR register.

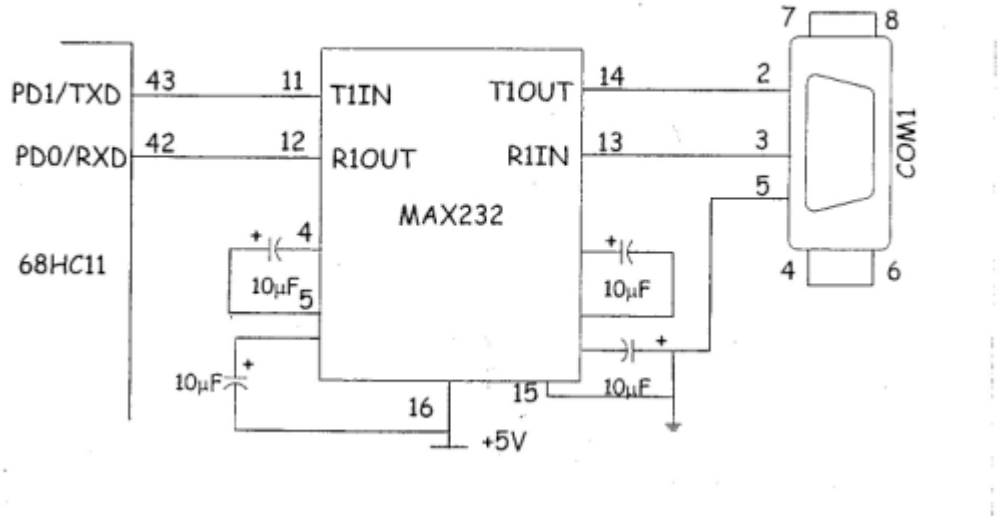
**Bit 4** (IDLE) – when set to 1 it show that RXD line located in idle level.

**Bit 3, 2 & 1** (OR, NF, FE) – when set to 1 it show that have error happen in the serial reception data process.

### 6.4.1 Serial Communication Interface (SCI) (cont…)

**Example:** A 68HC11 microcontroller system need to connect to one PC through COM1 serial communication port. Show the connection circuit and write a program to execute:
-sent character string "satu" to PC when character '1' accepted from SCI.
-sent character string "rosbi" to PC when character '2' accepted from SCI.
-ignore other character.

**Solution:**
Serial signal that in/out from 68HC11 pin RXD & TXD is logic TTL (0 @ 5V). While serial signal that out/in from pin COM1 at PC is logic RS232 (+12V @ -12V). Need to use the chip driver or RS232 receiver as MAX232 and so on. Software should be started SCI registers. Select Baud 9600 rate and data 8 bit.



## 6.4.1 Serial Communication Interface (SCI) (cont…)

**Solution (program):**

```
* ATURCARA CONTOH SCI UTK SEM4233
* SINOPSIS:
*   BACA 1 AKSARA DARI SCI,
*      JIKA '1' CETAK 'SATU'
*      JIKA '2' CETAK 'ROSBI'
*      JIKA BUKAN '1' @ '2' ABAIKAN
*   ULANG
* PENGATURCARA: DR. ROSBI MAMAT
* TARIKH: 31/08/2000

* ANGGAP OFSET BAGI SCDR,SCCR1,SCCR2,SCSR,BAUD TELAH DITAKRIF

          LDX    #$1000       ;DAFTAR I/O BERMULA DI $1000
          LDAA   #$30
          STAA   BAUD,X       ;TETAPKAN K.BAUD=9600
          BCLR   SCCR1,X $10  ;FORMAT 1,8,1
          BSET   SCCR2,X $0C  ;BOLEHKAN HANTAR & TERIMA

ULANG     BSR    BACA1AKS     ;BACA 1 AKSARA DARI SCI. A=AKS
          CMPA   #'1'         ;AKSARA = '1' ?
          BEQ    CETAK1       ;YA! PERGI HANTAR 'SATU'
          CMPA   #'2'         ;AKSARA = '2' ?
          BEQ    CETAK2       ;YA! PERGI HANTAR 'ROSBI'
          BRA    ULANG        ;BUKAN '1' @ '2' - ULANG

CETAK1    LDY    #MESEJ1      ; TUDING Y KPD 'SATU'
          BSR    KEPC         ; HANTAR KE PC
          JMP    ULANG        ; BUAT LAGI
CETAK2    LDY    #MESEJ2      ; TUDING Y KPD 'ROSBI'
          BSR    KEPC         ; HANTAR KE PC
          JMP    ULANG        ; BUAT LAGI

MESEJ1    FCB   'SATU',0
MESEJ2    FCB   'ROSBI',0
```

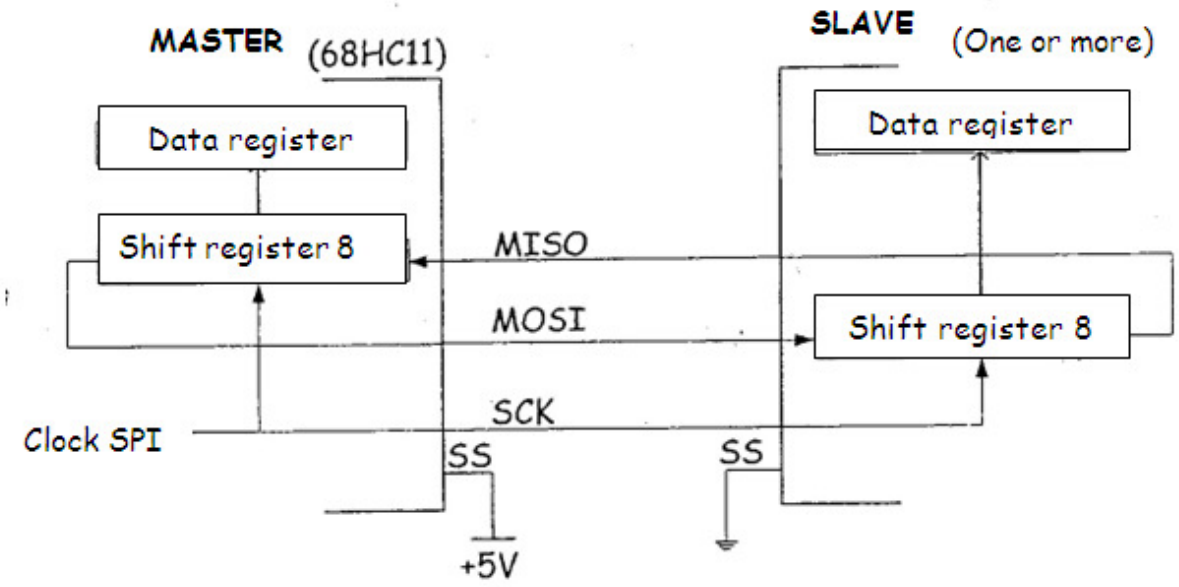
## 6.4.1 Serial Communication Interface (SCI) (cont…)

**Solution (program): …cont:**

```
*** SUBRUTIN2 YG GUNA SCI UTK HANTAR/TERIMA DATA
*** BOLEH DIGUNAKAN DLM ATURCARA LAIN

*BACA 1 AKSARA DARI SCI. PULANGKAN AKSARA DLM A

BACA1AKS  BRCLR  SCSR,X $20 BACA1AKS ; UJI BENDERA RDRF
          LDAA   SCDR,X       ;DAH ADA DATA, BACA DARI SCDR
          RTS                 ;BALIK KAMPUNG

*TULIS 1 AKSARA DLM A KE SCI.

TUL1AKS   BRCLR  SCSR,X $80 TUL1AKS  ; UJI BENDERA TDRE
          STAA   SCDR,X       ; DATA SEDIA UTK DIHANTAR
          RTS                 ;BALIK KAMPUNG


*HANTAR RENTETAN AKSARA KE PC MELALUI SCI-COM1
*ALAMAT RENTETAN DITUDING OLEH INDEKS Y
*RENTETAN MESTI DITAMATKAN OLEH SATU BAIT SIFAR

KEPC      LDAA   0,Y          ; CAPAI SATU AKSARA
          BEQ    DAHABIS      ; RENTETAN DAH TAMAT?
          JSR    TUL1AKS      ; HANTAR 1 BAIT KE SCI
          INY                 ; KEMASKINI PENUDING
          BRA    KEPC         ; ULANG HINGGA TAMAT RENTETAN
DAHABIS   RTS                 ; SELAMAT PULANG
```

### 6.4.2 Serial Peripheral Interface (SPI)

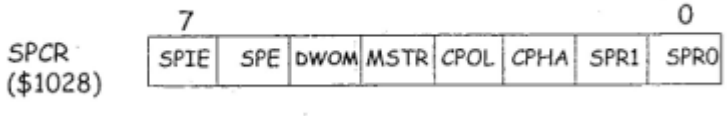
- It is a synchronous data communication.
- Use 4 pins from port D:
  1) MISO/PD2 – Master In Slave Out
  2) MOSI/PD3 – Master Out Slave In
  3) SCK/PD4 – Serial clock signal to sync data transfer.
  4) SS/PD5 – used to select slave device.
- Connection in the SPI is in the master-slave configuration.
  Master – usually is 68HC11. Data transfer started by master. SPI serial clock also produced by the master.
  Slave – consist of peripheral I/O SPI or 68HC11. Transfer the data as reaction from the master.




- SPI can transfer data 8-bit at a time.

### 6.4.2 Serial Peripheral Interface (SPI)

- Registers that involved in the SPI:
  1) SPDR ($102A) – data register for SPI. Transfer will be started when write to SPDR. In the same time data that shifted in stored in SPDR.
  2) SPCR ($1028) – register SPI controller. Use it to enable the SPI and select SPI clock rate.
  3) SPSR ($1029) – register data transfer status and SPI error.



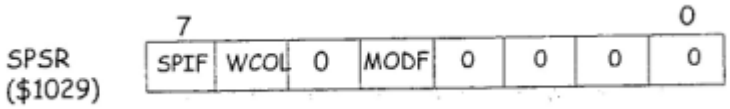

SPIE – set it to enable SPI interrupt.
SPE – set it to enable SPI operation.
DWON – delete it to produce CMOS output.
MSTR – if set SPI operate in the Master mode. '0' = slave
CPOL, CPHA – select polarity and SPI clock phase. '01' selects the negative clock trigger edge.
SPR1, SPR0 – select clock (SCK) rate to SPI.

| SPR1 | SPR0 | E clock divided |
|---|---|---|
| 0 | 0 | 2 |
| 0 | 1 | 4 |
| 1 | 0 | 16 |
| 1 | 1 | 32 |

- **Example**: calculate the faster and slower rate data transfer through SPI if crystal 8MHz is used.
- E clock = 8MHz/4 = 2MHz.
  Faster rate when SPR1, SPR0 = 00.
  Rate = 2MHz/2 = 1MHz = 1Mbit per second
  Slower rate when SPR1, SPR0 = 11.
  Rate = 2MHz/32 = 62.5KHz = 62.5Kbit per second

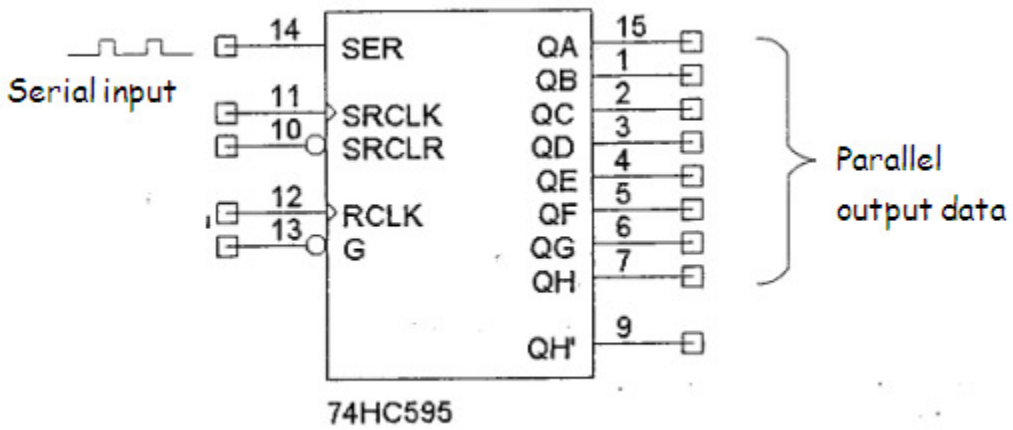
### 6.4.2 Serial Peripheral Interface (SPI)

- Registers that involved in the SPI:



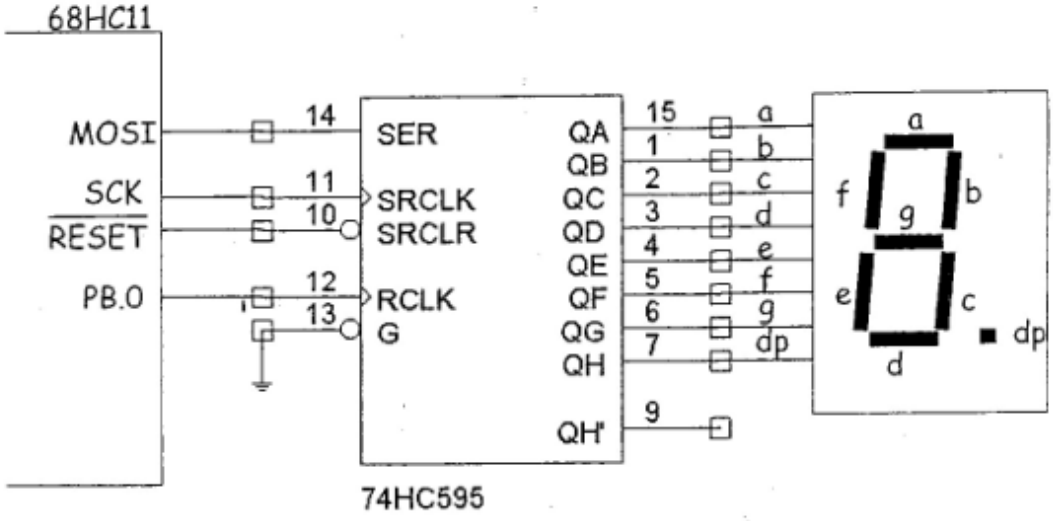

SPIF – transfer completion flag. Set it when data transfer through SPI is completed.
WCOL – set it when data written to SPDR when data transfer is happen.
MODF – when set indicates an error which more than one master in SPI connection network.

- There are many compatible chip interface with SPI protocol:
  MC14499 – LED 7 segments drivers and decoders display 4 digit
  MC145051 – 10 bit A/D converter
  MC144110 – 6 bit A/D converter
  MC68HC68T1 – real-time clock chip
  74HC595 – shift register 8 bit with serial input parallel output
  74HC589 – shift register 8 bit with parallel input serial output
- Example: How to produce a port 8 bit output with 74HC595 to connect a LED 7 segments display through SPI.



### 6.4.2 Serial Peripheral Interface (SPI)

- Example: …
  SER – serial input data
  SRCLK – data at SER pin will be shifted into shift register every positive trigger edge at SRCLK clock signal
  SRCLR – reset signal for shift register. Active low
  RCLK – positive trigger edge clock signal will be latch shift register content at serial output (QA-QH)
  G – signal to enable QA-QH output. Active low




```
        LDX     #$1000
        LDAA    #%00111000  ;SS,SCK,MOSI= OUTPUT
        STAA    DDRD,X
        LDAA    #%01010100  ;BOLEHKAN SPI SBG TUAN
        STAA    SPCR,X
        BSET    PORTB,X $01 ;RCLK =1

LAGI    LDAA    #%11000000  ;PAPARKAN SIFAR
        STAA    SPDR,X          ;HANTAR KE SPI
TUNGGU  LDAA    SPSR,X          ;TUNGGU SEHINGGA 8
BIT
        BPL     TUNGGU          ;SELESAI DI ANJAK

        BCLR    PORTB,X $01 ;HASILKAN ISYARAT RCLK
        BSET    PORTB,X $01 ;PICUAN PINGGIR +VE
        JMP     LAGI
```