Our Products Make Your Product Better®

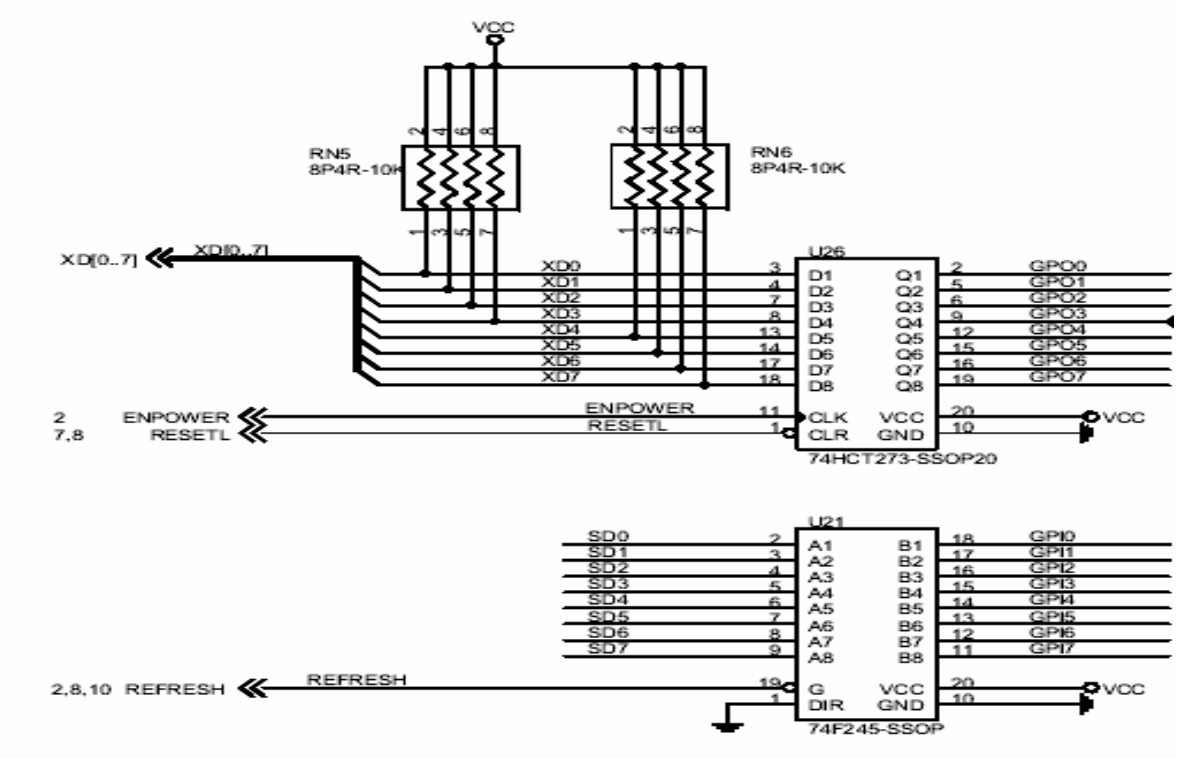
To learn more about EMAC's products and services and how they can help your project

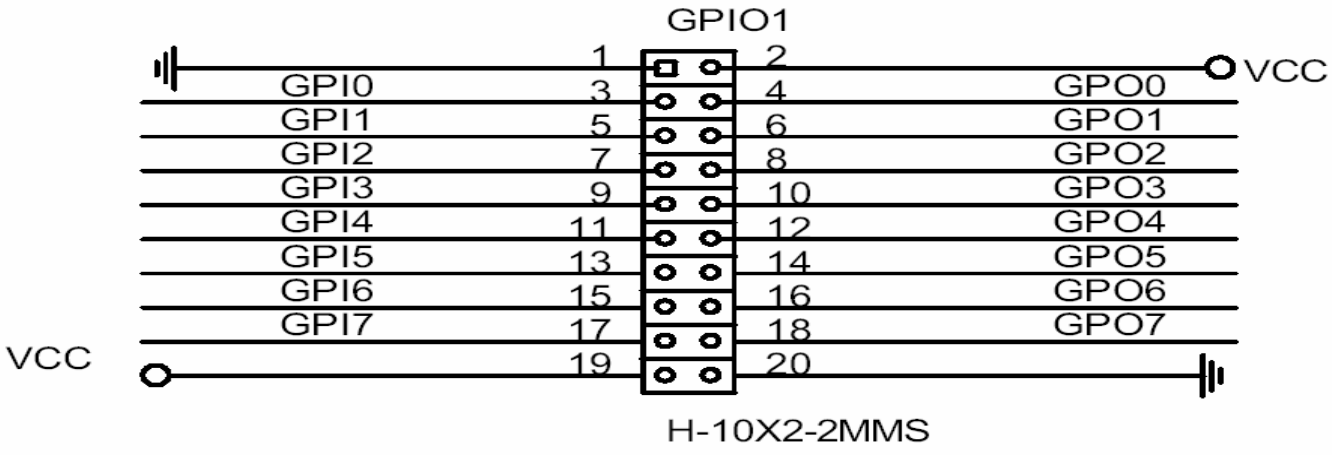
http://ftp.emacinc.com/Tech_Info/About_EMAC_Products_and_Services.pdf

Authorized Distributor, Integrator, and Value-Added Reseller

GPIO for ALi 6117

M6117 supports 8 expandable GPOs and 8 expandable GPIs. During normal condition, pins XD[7:0] are data bus to peripheral devices. But during cold reset, XD[7:0] is an input pin and latched by internal register - index 68h; the pin ENPOWER is also active at this time to latch XD[7:0] at external 74LS373. Because there is no default value in index 68h and as to XD[7:0] without any pulling resistor. Designer has to connect externally pull-up or pull down resistors to XD[7-0] to initialize index 68h. The index 68h : D[7-0] are both readable and writable. If BIOS wants to change the external 74LS373 latch value. It should first set index 68h :D[7:0] a new value, then write any value to index 73h, that will generate an ENPOWER signal to update 74LS373 latch value. The index value in 68h will appear at XD[7:0] bus and ENPOWER will update the XD value to 74LS373. [Index 3Eh: The low byte GPI value. Default 00h Read only.]





1. Generate GPOs method

(1) Use external 74373 input connect to SD bus. The latch enable pin connects to ENPOWER.
(2) Set index 68h to desired GPO value.
(3) Write index 73h.
(4) Then data stored in index 68h will be sent to SD[7:0] and XD[7:0]. and ENPOWER will be active.
(5) The value will be latched by 74373.

2. Generate GPIs method

(1) Add external 74245, the input connects to GPIs, the output connects to ISA SD bus. The OE control connects to ISA REFRESHJ.
(2) When REFRESHJ is active, the SD will become input and M6117 will use MEMRJ rising edge to latch the SD value.
(3) Every 15us, the GPIs value will be updated.
(4) It can read the GPI value through index 3Eh which store SD[7:0] value.

※ Configuration Registers

1. Register Bit Definition:

The details of M6117 configuration registers are described as follows :

PORT 22H	default 00H
Bit	Description
7~0	Index of Configuration register
PORT 23H	default 00H
Bit	Description
7~0	Data of Configuration register if unlock register unlocked.

INDEX 3EH	default 00H
Bit	Description
7~0	GPI signals. When REFRESHJ is active, the SD will become input and M6117D will use MEMRJ rising edge to latch the SD[7:0] to Bit 7~0. Read only
INDEX 68H	default 00H
Bit	Description
7~0	Power ON latched Power Control Initial status from XXD[7-0] D[7-0] : PWR[7-0] control pin status
INDEX 73H	default 00H
Bit	Description
7~0	Power Control status output command Write to this port will generate enpower pulse to update power control status.

2. How to read/write to configuration registers

The read/write configuration register is the first index to be processed. On board I/O port 22h is the index register and I/O port 23h is the data register. To read a configuration register, write the index value to I/O port 22h in advance, then read data from I/O port 23h. To write a configuration register, write the index value to I/O port 22h, then write data to I/O port 23h. For instance, if we want to read the data of configuration register which index is 10h, the steps are :

1) Write 10h (index) to I/O port 22h
2) Read data from I/O port 23h

If we want to write data 55h to configuration register which index is 12h, then the steps are :

1) Write 12h (index) to I/O port 22h
2) Write data 55h to I/O port 23h

*The steps of locking/unlocking the configuration registers :

```
OUT 22h, 13h      (Enable 13h)
OUT 23h, C5h      (Unlock)
OUT 22h, XXh      (XX = Configuration Index)
OUT 23h, YYh      (YY = Configuration data)
OUT 22h, XXh
OUT 23h, YYh (Configuration can be written repeatedly)
      :
OUT 22h, 13h      (Enable 13h)
OUT 23h, 00h      (Lock)
```

※ Programming Guide - Basic Procedure and Macro Definition

a) Delay

```
IO_Delay MACRO
        jcxz$+2
        jcxz$+2
        ENDM
```

b) Unlock chipset configure registers

```
Open_Chip MACRO
        mov     al, 013h
        out  022h, al
        IO_Delay
        mov     al, 0c5h
```

```
        out  023h, al
        IO_Delay
        ENDM
```

c) Lock chipset configure registers

```
Close_Chip MACRO
        mov     al, 013h
        out  022h, al
        IO_Delay
        mov     al, 000h
        out  023h, al
        IO_Delay
        ENDM
```

d) Write data to configure register

```
; INPUT : AH  -      INDEX#
; INPUT : AL  -      Data
; ACTION :      Write the value of AL into the value of AH INDEX
; Interrupt controller and Stack are available
Write_To_Chip PROCEDURE
        cli
        push    ax
        Open_Chip
        pop ax
        out  022h, al
        IO_Delay
        xchg    ah, al
        out  023h, al
        IO_Delay
        xchg    ah, al
        push    ax
        Close_Chip
        pop ax
        sti
        ret
        ENDP
```

e) Read data from configure register

```
; INPUT : AL  -      INDEX#
; OUTPUT :     AL  -      Data
; ACTION :      Read data from the value of AL INDEX
```

```
; Interrupt controller and Stack are available
Read_From_Chip PROC
        cli
        push    ax
        Open_Chip
        pop ax
        out  022h, al
        IO_Delay
        in    al, 023h
        IO_Delay
        push    ax
        Close_Chip
        pop ax
        sti
        ret
        ENDP
```