Embedding it better...

µTasker Document

M5214, M5216, M5280, M5281, M5282

Table of Contents

1. Introduction

The µTasker project for the Coldfire V2 started with the availability of the M5223X. This device is a true single chip processor with embedded Ethernet FEC and PHY. The M5282 (and others belonging to this family of high pin count BGA devices) are older devices with internal FEC (requiring external PHY) but are also true single chip processors since they all do have internal RAM and (apart for the M5280) FLASH.

A quick comparison between the M5223X and these reveals the following main differences:

- The internal FEC requires an external PHY to complete the Ethernet interface
- The devices are available exclusively in 256 MAPBGA
- 2k-byte cache
- The ADC is known as QADC, which is different to the version in the M5223X
- External bus interface supporting SDRAM and the ability to boot from an external device; up to 32 data bus width and 24 bit address bus and up to 7 chip select lines
- Two 4-channel general purpose timers (instead of one) but no PWM module
- Larger quantity of ports, possibly shared with the external bus interface

The following document concentrates on the most important details for understanding how to use the M5282, using the M5223X project as comparison. It does not attempt to replace the user's guide but instead tries to keep things as simple as possible – identifying and elaborating on the most important practical points.

The device name M5282 is used generally for all of the following devices:

- M5214 – 64k SRAM and 256k FLASH; CAN but no Ethernet
- M5216 – 64k SRAM and 512k FLASH; CAN but no Ethernet
- M5280 – 64k SRAM but no FLASH
- M5281 – 64k SRAM and 256k FLASH
- M5282 – 64k SRAM and 512k FLASH

2. Processor Modes

The M5282 operates in one of the following modes: MASTER MODE or SINGLE-CHIP MODE.

The configuration is chosen at reset and cannot be changed afterwards.

- MASTER MODE allows external memory interface operation, whereby the width of the interface is configurable and can be programmed to a certain degree by configuring ports.
- SINGLE_CHIP MODE operates without any external memory interface and all of the ports are general purpose. These cannot be changed to operate as external memory interface.

The Chip Configuration Module (CCM) is responsible for the configuration.

If the control input RCON is pulled high (‘1’) at reset the chip will enter SINGLE-CHIP MODE with all ports configured for full drive strength, otherwise further inputs are read to determine the configuration details.

D16, D17 and D26 select one of the two modes:

Data lines read at reset when RCON is asserted	Port bits	MASTER MODE	SINGLE_CHIP
D16	PORT_B_BIT0	‘1’	‘0’
D17	PORT_B_BIT1	‘1’	‘1’
D18	PORT_B_BIT2	Boot device	Boot device
D19	PORT_B_BIT3	Boot device	Boot device
D21	PORT_B_BIT5	Port drive strength	Port drive strength
D24	PORT_A_BIT0	Chip select config.	Chip select config.
D25	PORT_A_BIT1	Chip select config.	Chip select config.
D26	PORT_A_BIT2	‘1’	‘1’

The boot device configuration controls the CS0 line to be asserted as 8-bit, 16-bit or 32-bit port when D18 or D19 are not both ‘0’.

D19 (PORT_B_BIT3)	D18 (PORT_B_BIT2)	Boot width
0	0	Internal 32 bit – CS0 not asserted
0	1	16-bit width
1	0	8-bit width
1	1	32-bit width

When D21 is read as ‘1’ the port drive strength is set to full drive strength. ‘0’ results in partial strength.

The chip select configuration controls the A[23:21] and CS[6:4] lines

D25 (PORT_A_BIT1)	D24 (PORT_A_BIT0)	PF[7]	PF[6]	PF[5]
0	0	A[23]	A[22]	A[21]
0	1	CS[6]	A[22]	A[21]
1	0	CS[6]	CS[5]	A[21]
1	1	CS[6]	CS[5]	CS[4]

The configuration lines need to be controlled by external hardware during reset.

When simulating the M5282 using the µTasker simulator the mode can be controlled in `app_hw_m5223x.h`. By setting the define `CHIP_CONFIGURATION` accordingly.

```
#define CHIP_CONFIGURATION RCON_ASSERTED  // read configuration from ports
```

or

```
#define CHIP_CONFIGURATION         0      // default - single-chip mode
```

The initial input state can furthermore be defined by setting the ports A and B:

```
#define A_DEFAULT_INPUT            0xfc
#define B_DEFAULT_INPUT            0xf3
```

This corresponds to a configuration for master mode, partial drive strength, internal boot and A[23:A21].

The values of `CCM_RCON` and `CCM_CCR` after initialization can be checked in the simulator to verify that the settings correspond to the desired mode. In addition, the default setting of various ports are conditional on the configuration.

Eg. To check the value of `CCM_RCON` in the µTasker simulator do the following:

1) Search for the register name in the file `M5223x.h`. The define is as follows:

   ```
   #define CCM_RCON    *(unsigned short *)(RESET_CTRL_ADD + 0x08)
   ```

   This shows that the register belongs to the `RESET_CTR_ADD` block. It is 16 bits wide and located at an offset of 0x08 from the start of the block.

2) Now search for the definition of `RESET_CTR_ADD`. It will be found twice, once for hardware use and once for simulator use:

   ```
   #define RESET_CTRL_ADD      (IPSBAR + 0x110000)

   #define RESET_CTRL_ADD      ((unsigned char *)(&ucM5223X.SimRESET))
   ```

3) Select the struct `ucM5223X.SimRESET` and drag it into the simulator's watch window.
4) In the watch windows the struct can be expanded and the register contents displayed. Note that this can only be performed when the simulator has been paused.

All other registers can be viewed in an analog manner.

3. Clock Modes

The clock mode is selected during reset by the values applied to the pins CLKMOD[1:0]. The value set at reset cannot be changed.

The value selected is reflected by several bits in the MCF_CLOCK_SYNSR register.

CLKMOD[1]	CLKMOD[0]	Clock mode
0	0	External clock mode (PLL disabled)
0	1	1:1 PLL mode
1	0	Normal PLL mode, external clock reference
1	1	Normal PLL mode, crystal clock reference

Normally the mode [11] is used since the software can configure the PLL to achieve the operating speed that is required.

4. Ports

One of the largest differences compared with the M522XX is the fact that the M5282 has a large number of ports. Some of these are shared with the external memory interface and some have fixed operating modes depending on the operating mode of the processor (MASTER MODE or SINGLE_CHIP MODE).

The ports can be divided into three groups. The first is configured depending on the processor's operating mode and cannot all be used as GPIO when operating in master mode. The second group can be fully controlled via their configuration registers and default to GPIO function (apart from Port DD when the BDM interface is active). The thirst group does not belong to the GPIO module and the individual pins are controlled by their peripheral.

Configuration dependent ports	GPIOs	Peripherals
Port A - always D31..24 in master mode. Always Port A in single-chip mode	Port AS - defaults to GPIO	Port NQ - *Edge port module*
Port B - defaults to D23..D16 in master mode when the bus with is 32 bits. *Can be modified*	Port QS - defaults to GPIO	Port QA – *QADC module*
Port C - defaults to D15..D8 in master mode when the bus with is 16 or 32 bits. *Can be modified together with Port D*	Port TC - defaults to GPIO	Port QB – *QADC module*
Port D - defaults to D7..D0 in master mode when the bus with is 16 or 32 bits. *Can be modified together with Port C*	Port TD - defaults to GPIO	Port TA – *GPT0 module*
Port E - defaults to external memory interface signals in master mode. *Can be modified*	Port UA - defaults to GPIO	Port TB – *GPT1 module*
Port F- F[7:5] default to external memory interface signals in master mode. *Can be modified* F[4:0] are always A[20:16] in master mode	Port DD – defaults to primary function (DDATA[3:0] and PST[3:0]). *Controlled in CCM_CCR*	
Port G - always A15..8 in master mode. Always Port G in single-chip mode	Port EH (not M5214/M5216) - defaults to GPIO	
Port H - always A7..0 in master mode. Always Port H in single-chip mode	Port EL - defaults to GPIO	
Port J - defaults to external memory interface signals in master mode. *Can be modified*		
Port SD - defaults to external memory interface signals in master mode. *Can be modified*		

Note that, for comparison, all GPIO of the M522XX would belong to the GPIO group.

PE[3:2] default to SIZ[1:0] in MASTER-MODE. These can be disabled to GPIO pins by clearing CCR_TSIZ in CCM_CCR and also clearing the pin function in PEPAR:

```
CCM_CCR &= ~CCR_TSIZ;  // disable SIZ[1:0] peripheral function
PEPAR &= ~(PEPAR_PEBIT2 | PEPAR_PEBIT3); // set PE[3:2] (SIZ[1:0]) to GPIO
```

The GPIO pins (apart from the port DD) all default to GPIO operation. To use peripherals on these ports the peripheral function is programmed in the port's pin assignment register. This programming is performed automatically when µTasker peripheral drivers are used – in some cases the peripheral pins can be assigned to multiple possible locations, whereby the setting to be used is configured in this case in the file app_hw_m5223X.h.

Peripheral ports do not have their own pin configuration register and so the function is controlled directly by the corresponding peripheral module as explained below:

Port NQ

This 7-pin port belongs to the Edge Port Module (EPORT). It has 7 external interrupt pins (IRQ1..IRQ7). The port defaults to GPIO input whereby the port state is read by reading EPPDR0. The port can be used as an output by configuring with EPDDR0.

The µTasker simulator displays this pin as peripheral function as long as it is programmed as an input (either with or without active interrupts enabled). When a pin is programmed as an output (corresponding EPDDR0 bit is sent and state controlled by EPDR0) it is displayed as an output.

When the inputs are toggled in the µTasker simulator a matching interrupt (level or edge sensitive) will invoke the corresponding edge port interrupt to be executed.

Note that INT7 has NMI (non-maskable interrupt) interrupt level and its use should be considered carefully since it can interrupt any other interrupt presently being processed.

Ports QA and QB

Ports QA and QB belong to the Queued Analogue-to-Digital Converter (QADC). They are both 4 bit ports, whereby QA has pins QA[4:3] and QA[1:0] and QB has QB[3:0].

By default these ports are digital inputs. The QADC module also has a set of port registers: PORTQA, PORTQB, DDRQA and DDRQB. When the QADC is configured and uses these pins for analog or multiplex control functions the peripheral mode is automatically selected in preference to the GPIO mode.

At the time of writing the µTasker project doesn't support the QADC module and so these ports are simulated in the µTasker simulator as pure GPIOs.

Ports TA and TB

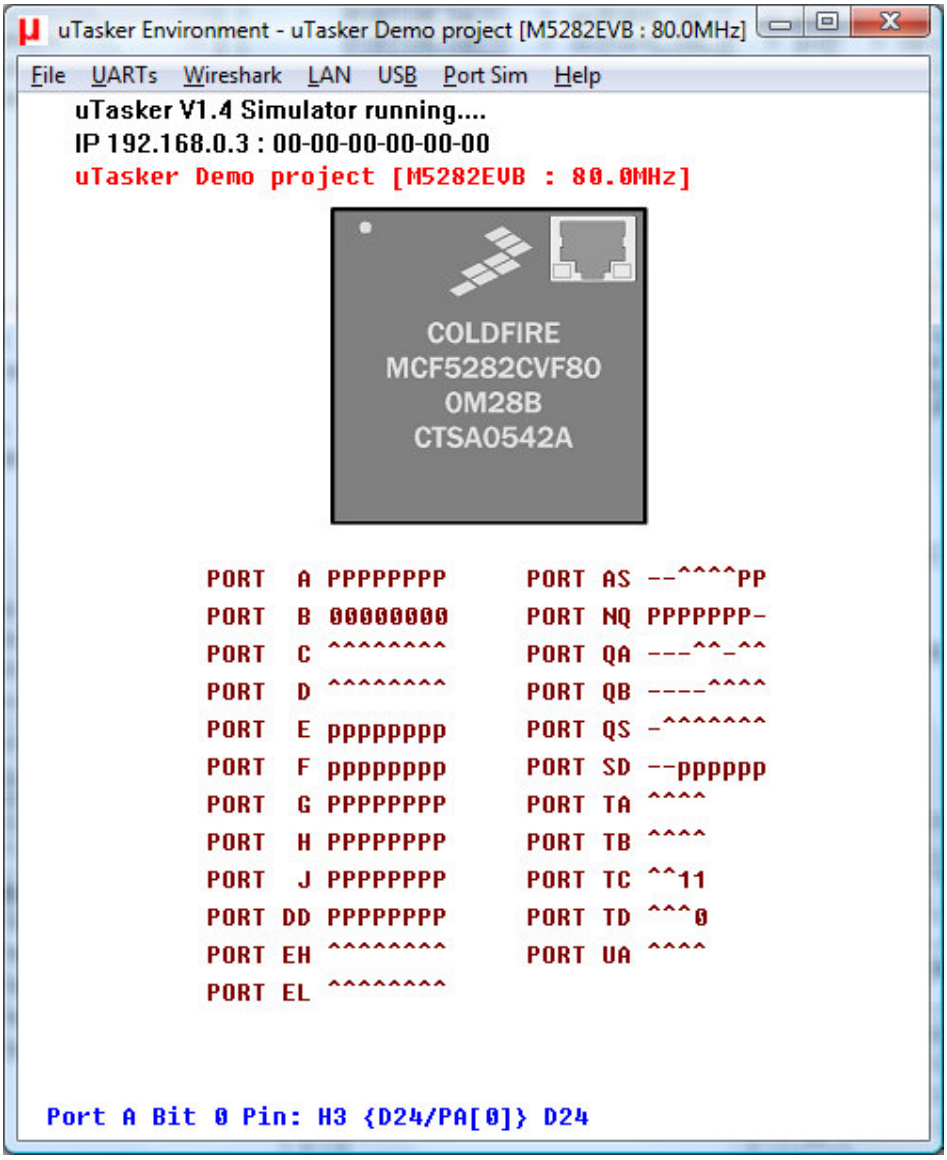
Ports TA and TB belong to the two 4-channel general purpose timer modules GPT0 and GPT1.

The M522XX, for comparison, has one 4-channel general purpose timer module and a PWM module instead of the second one.

By default these ports are digital inputs. The GPT modules have a set of port registers: `GPTPORT`, `GPT1PORT`, `GPTDDR` and `GPT1DDR`. When the GPT is configured and uses these pins for GPT function the peripheral mode is automatically selected in preference of the GPIO mode.

At the time of writing the µTasker project doesn't support the GPT pin control and so these ports are simulated in the µTasker simulator as pure GPIOs.

The µTasker simulator shows the ports, including their peripheral function and state as it is operating, making verification of their configuration possible. By hovering the mouse over the port of interest its possible configuration as well as actual function is displayed as well as the pin number where the connection can be found on the package.



µTasker simulator displaying the M5282 and its ports during operation

5. Peripheral Power Management

The M5282 doesn't have any registers for controlling the peripheral module clocks as the M522XX has.

6. FEC

The M5282EVB uses the AMD Am79C875VC as PHY. The PHY has the address 0x01 and reads the identifier code 0x0022561b (whereby the last 4 bits are the chip's revision number).

When the Ethernet interface is enabled port EH and EL are configured for peripheral use – these are the MII lines between the FEC and the PHY. Port AS[5:4] are used as management interface to the PHY (EMDIO and EMDC).

7. UARTs

The M5282 has 3 UARTs. UARTs 0 and 1 have RTS and CTS lines. UART2 doesn't have dedicated RTS/CTS signals.

UART Signal	Port and Pin location	Alternative 1	Alternative 2	Alternative 3
UTXD0	PUA[0] – Pin T7			
URXD0	PUA[1] – Pin N6			
UCTS0	PTC[0] – Pin K13	PTC[1] – Pin K14	PTD[0] – Pin J13	PTD[1] – Pin J14
URTS0	PTC[2] – Pin K15	PTC[3] – Pin K16	PTD[2] – Pin J15	PTD[3] – Pin J15

UART Signal	Port and Pin location	Alternative 1	Alternative 2	Alternative 3
UTXD1	PUA[2] – Pin P7			
URXD1	PUA[3] – Pin R7			
UCTS1	PTC[0] – Pin K13	PTC[1] – Pin K14	PTD[0] – Pin J13	PTD[1] – Pin J14
URTS1	PTC[2] – Pin K15	PTC[3] – Pin K16	PTD[2] – Pin J15	PTD[3] – Pin J15

UART Signal	Port and Pin location	Alternative 1	Alternative 2
UTXD2 (M5214 and M5216)	PAS[0] – Pin E15	PAS[2] – Pin E13	**PAS**[4] – Pin A10
UTXD2 (M528X)	PAS[0] – Pin E15	PAS[2] – Pin E13	
URXD2	PAS[1] – Pin E14	PAS[3] – Pin D16	PAS[5] – Pin C10

To configure the use of alternative pins for UTXD2 and URXD2 the following defines can be activated in `app_hw_m5223x.h`:

`UART2_TXON_AS_4` (only M5214 and M5216)

`UART2_TXON_AS_2`

`UART2_RXON_AS_5`

`UART2_RXON_AS_3`

8. Cache

Open.

9. External Interface and Chip Select Modules

Open.

10. SDRAM Controller Module

Open.

11. Conclusion

This document is in progress and has not been officially released.

Modifications:
- V0.01 26.4.2010 – provisional version for the documentation page for development monitoring

- V0.02 28.4.2010 – provisional version for the documentation page for development monitoring. Added Peripheral ports.

- V0.03 29.4.2010 – provisional version for the documentation page for development monitoring. Added Peripheral Power Management, FEC and UARTs.