Introduction

Memory Map, Communication, Polling and Interrupts

Exercise class 1

Presenter:
Jürgen Mattheis

In cooperation with:
Pascal Walter

Based on the lecture of:
Marco Zimmerling

November 15, 2022

University of Freiburg, Chair for Embedded Systems

Gliederung

About this exercise class

About this exercise class

- based on the new lecture by Prof. Zimmerling
- the exercise class gets recorded and streamed online and is also hold in presence
- feedback for me: `https://forms.gle/f3YN8EFrZ1vsfPoC6`
- today a little preview of the exercises that await you
- solving the exercises not directly necessary for the Studienleistung, important for exam $\Rightarrow$ by passing the exam you're also going to get your Studienleistung

Task 1 - Memory map

Task 1 - Memory Map

Task 1.1

Solution a:

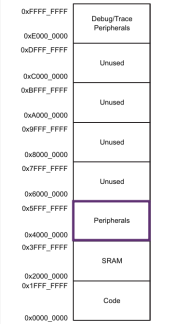


Figure 6-1. Device Memory Zones

- $0x5FFF\_FFFF - 0x4000\_0000 + 1 = 0x2000\_0000$
- $0x2000\_0000 = 2 \cdot 16^7 = 2 \cdot (2^4)^7 = 2 \cdot 2^{4 \cdot 7} = 2^{1+28} = 2^{29}$

Task 1 - Memory Map

Task 1.1

Solution b:

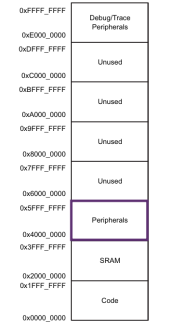


Figure 6-1. Device Memory Zones

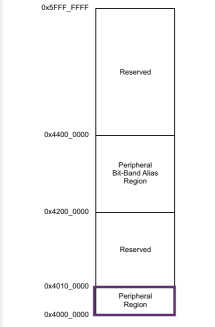


Figure 6-4. Peripheral Zone Memory Map

6.3.3.1 Peripheral Region

The 1MB region from 0x4000_0000 to 0x400F_FFFF is dedicated to the system and application control peripherals of the device. On the MSP432P401x MCUs, a total of 128KB of this region is dedicated for peripherals, while the rest is reserved. Table 6-1 lists the peripheral allocation within this 128-KB space. Note that all peripherals may not be available in all devices of the family (details in the REMARKS column). If a peripheral is listed as N/A for a particular device, treat the corresponding address space as reserved.

ADDRESS RANGE	PERIPHERAL	TABLE	REMARKS
0x4000_2800 to 0x4000_2BFF	eUSCI_B2	Table 6-12	16-bit peripheral
0x4000_2C00 to 0x4000_2FFF	eUSCI_B3	Table 6-13	16-bit peripheral
0x4000_3000 to 0x4000_33FF	REF_A	Table 6-14	16-bit peripheral
0x4000_3400 to 0x4000_37FF	COMP_E0	Table 6-15	16-bit peripheral
0x4000_3800 to 0x4000_3BFF	COMP_E1	Table 6-16	16-bit peripheral
0x4000_3C00 to 0x4000_3FFF	AES256	Table 6-17	16-bit peripheral
0x4000_4000 to 0x4000_43FF	CRC32	Table 6-18	16-bit peripheral
0x4000_4400 to 0x4000_47FF	RTC_C	Table 6-19	16-bit peripheral
0x4000_4800 to 0x4000_4BFF	WDT_A	Table 6-20	16-bit peripheral
0x4000_4C00 to 0x4000_4FFF	Port Module	Table 6-21	16-bit peripheral
0x4000_5000 to 0x4000_53FF	Port Mapping Controller	Table 6-22	16-bit peripheral

- $0x4000\_4FFF - 0x4000\_4C00 + 1 = 0x0400$
- $0x0400 = 4 \cdot 16^2 = 2^2 \cdot (2^4)^2 = 2^2 \cdot 2^{(4 \cdot 2)} = 2^{2+8} = 2^{10}$

Task 1 - Memory Map I

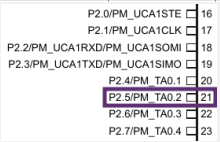
Task 1.1

Solution c:

ADDRESS RANGE	PERIPHERAL	TABLE	REMARKS
0x4000_2B00 to 0x4000_2BFF	eUSCI_B2	Table 6-12	16-bit peripheral
0x4000_2C00 to 0x4000_2FFF	eUSCI_B3	Table 6-13	16-bit peripheral
0x4000_3000 to 0x4000_33FF	REF_A	Table 6-14	16-bit peripheral
0x4000_3400 to 0x4000_37FF	COMP_E0	Table 6-15	16-bit peripheral
0x4000_3800 to 0x4000_3BFF	COMP_E1	Table 6-16	16-bit peripheral
0x4000_3C00 to 0x4000_3FFF	AES256	Table 6-17	16-bit peripheral
0x4000_4000 to 0x4000_43FF	CRC32	Table 6-18	16-bit peripheral
0x4000_4400 to 0x4000_47FF	RTC_C	Table 6-19	16-bit peripheral
0x4000_4800 to 0x4000_4BFF	WDT_A	Table 6-20	16-bit peripheral
0x4000_4C00 to 0x4000_4FFF	Port Module	Table 6-21	16-bit peripheral
0x4000_5000 to 0x4000_53FF	Port Mapping Controller	Table 6-22	16-bit peripheral

Table 6-21. Port Registers (Base Address: 0x4000_4C00)

REGISTER NAME	ACRONYM	OFFSET
Port 1 Input	P1IN	000h
Port 2 Input	P2IN	001h
Port 1 Output	P1OUT	002h
Port 2 Output	P2OUT	003h



- $0x4000\_4C00 + 0x0003 = 0x4000\_4C03$
- *6th least significant bit* $\Rightarrow 0b0010\_0000$

Task 1 - Memory Map II

Task 1.1

Sidenote

- 003h is an alternative way to write 0x0003

Task 1 - Memory Map I

Task 1.1

Solution d:

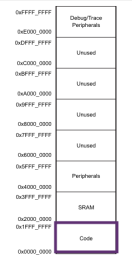


Figure 6-1. Device Memory Zones

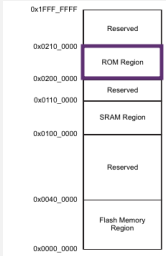


Figure 6-2. Code Zone Memory Map

6.3.1.1 Flash Memory Region

The 4MB region from 0x0000_0000 to 0x003F_FFFF is defined as the flash memory region. This region is further divided into different types of flash memory regions, which are explained in Section 6.4.1.

6.4.3 ROM

The MSP432P401x MCUs support 32KB of ROM, with the rest of the 1MB region reserved (for future upgrades). The lower 2KB of the ROM is reserved for TI internal purposes and accesses to this space returns an error response. The rest of the ROM is used for driver libraries.

NOTE

The entire ROM region returns an error response for write accesses. The lower 2KB of the ROM always returns an error response for any access.

- $0x020F\_FFFF - 0x0200\_0000 + 1 = 0x0010\_0000$.
- $16^5 = (2^4)^5 = 2^{4 \cdot 5} = 2^{20}$ *addresses*

Task 1 - Memory Map II

Task 1.1

Solution d:

- *each address location corresponds to one byte* $\Rightarrow$ *addressable memory space is* 2^{20}*Byte or* 1*MiB.*
- *number of 4-byte words is a quarter of that, which is* $\frac{2^{20}Byte}{2^2} = 2^{18}words = 2^8 Kiwords = 256 Kiwords$

Sidenote

- developer can not write to these addresses, only the manufacturer can (ROM = Read-only memory)
- the MSP432P401x MCUs supports 32KB of ROM, and the rest of the 1MByte ROM region is reserved for future upgrades

Task 1 - Memory Map

Task 1.2

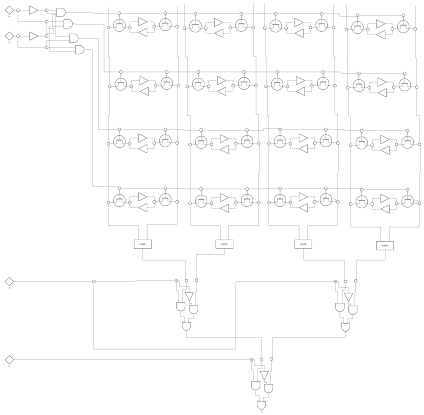


Figure 1: SRAM-cell array

Task 1 - Memory Map

Task 1.2

- u row select bits and w column select bits
- we need:
  - $2^{(u+w)}$ memory cells
  - one u-bit decoder
  - one 2^w-to-1 multiplexer
  - 2^w sense amplifiers

Task 1 - Memory Map I

Task 1.2

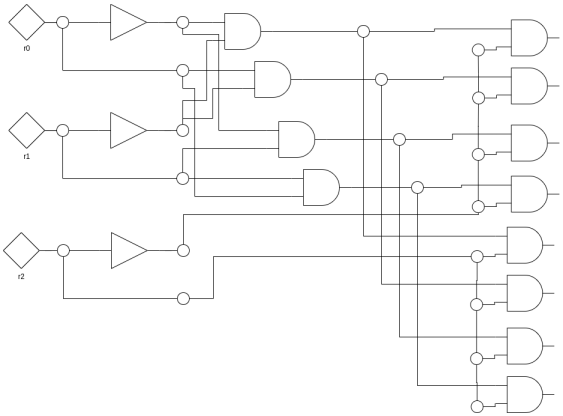


Figure 2: 3-Bit Decoder

Task 1 - Memory Map II

Task 1.2

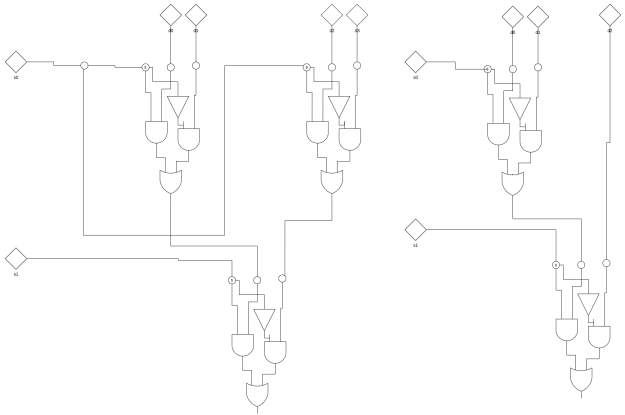


Figure 3: 4-to-1 and 3-to-1 Multiplexer

Task 1 - Memory Map I

Task 1.2

- memory cell area:
  - for a 8-bit address, we need 2^8 memory cells.
  - $C = 2^8 \cdot A_{\text{mem}} = 2^8 \cdot 6 = 1536$.
- u-bit decoder area:
  - for activating one of the 2^u word lines.
  - we construct a k-bit decoder using 2 smaller decoders and 2^k 2-input AND gates.
  - the smaller decoders should be of size $\frac{k}{2}$ if k is even, or of sizes $\frac{k+1}{2}$ and $\frac{k-1}{2}$ if k is odd.

Task 1 - Memory Map II

Task 1.2

- $$D(k) = \begin{cases} A_{\mathrm{NOT}} & \text{if } k = 1 \\ 2 \cdot D\left(\frac{k}{2}\right) + A_{\mathrm{AND}} \cdot 2^k & \text{if } k > 1 \text{ and } k \text{ is even} \\ D\left(\frac{k-1}{2}\right) + D\left(\frac{k+1}{2}\right) + A_{\mathrm{AND}} \cdot 2^k & \text{if } k > 1 \text{ and } k \text{ is odd} \end{cases}$$

- $2w$-to-1 multiplexer area:
  - for selecting one of the 2^w bit lines.
  - we can construct a 2^k-to-1 multiplexer for any k using two 2^{k-1}-to-1 multiplexers and one 2-to-1 multiplexer.
  - $$M(k) = \begin{cases} A_{\mathrm{mux}} = 4 & \text{if } k = 1 \\ 2 \cdot M(k-1) + A_{\mathrm{mux}} & \text{otherwise} \end{cases}.$$

Task 1 - Memory Map III

Task 1.2

- single sense amplifier area:
  - for each of the w column bit lines
  - $S(w) = 2^w \cdot A_{\text{sense}}$
- there are 8 possible implementations for u and w
- the area excluding the memory cells: D(u) + M (w) + S(w)

Task 1 - Memory Map

Task 1.2

u	w	Decoder $D(u)$	Multiplexer $M(w)$	Sense Amp $S(w)$	Total
0	8	0	1020	1280	2300
1	7	1	508	640	1149
2	6	6	252	320	578
3	5	15	124	160	299
4	4	28	60	80	168
5	3	53	28	40	121
6	2	94	12	20	126
7	1	171	4	10	185
8	0	312	0	5	317

Table 1: Different SRAM implementations

Task 2 - Communication

Task 2 - Communication

Task 2

Solution a:

- *Baudrate of* $115200 \frac{bits}{s}$
- *We require 16 clock periods to sample* $1bit$
- *Required clock frequency* $= 115200 \frac{bits}{s} \cdot \frac{16}{bits} = 1.8432MHz$

Task 2 - Communication

Task 2

Solution b:

- *We have a 48MHz clock and want to reduce it with division factor x*
- *Our goal speed is 1.8432 MHz*
- *The ration between the given and required clock is* $\frac{48MHz}{1.8432MHz} = 26.04167$

Task 2 - Communication

Task 2

Solution c:

- *For each byte of actual data we send: 1 start bit + 8 data bits + 1 parity bit + 2 stop bits = 12 bits*
- *We have a total of 10MB* $= 10 \cdot 2^{20}$ *bytes of data to send*
- *This takes* $\frac{(10 \cdot 2^{20} \cdot 12) bits}{115200 bits/s} = 1092.27s$ *time*

Task 2 - Communication I

Task 2

Solution d:

- *transmission rates sender and receiver with unknown clock frequency F_r:*
  - $r_s = \frac{48 \cdot 10^6}{16 \cdot 26}$ Hz, $\quad r_r = \frac{F_r}{16 \cdot 26}$
- *the sender transmits the kth symbol during time:*
  - $\left(\frac{(k-1)}{r_s}, \frac{k}{r_s} \right)$
- *receiver samples the kth symbol at time:*
  - $\frac{k-0.5}{r_r}$
- *For correct reception of the kth symbol, it has to be constant for a clock cycle before and after the sample time:*
  - $\left(\frac{k-0.5}{r_r} - \frac{1}{r_r \cdot 16}, \frac{k-0.5}{r_r} + \frac{1}{r_r \cdot 16} \right)$

Task 2 - Communication II

Solution d:

- *If receiver is slower than the sender, the second stop bit is 'critical'. To be sampled correctly, it has to be sampled before the second stop bit ends:*
  - $\frac{12-0.5}{r_r} + \frac{1}{r_r \cdot 16} \leq \frac{12}{r_s}$

$$r_r \geq \frac{11.5 \cdot 16 + 1}{12 \cdot 16} \cdot r_s = \frac{\frac{11.5 \cdot 16 + 1}{12 \cdot 16} \cdot 48 \cdot 10^6}{16 \cdot 26} \text{ Hz} = \boxed{\frac{46.25\text{MHz}}{16 \cdot 26}}$$

- *If receiver is faster than the sender, the first stop is 'critical'. To be sampled correctly, it has to be sampled after the start of the first stop bit:*
  - $\frac{11-0.5}{r_r} - \frac{1}{r_r \cdot 16} \geq \frac{10}{r_s}$

$$r_r \leq \frac{10.5 \cdot 16 - 1}{10 \cdot 16} \cdot r_s = \frac{\frac{10.5 \cdot 16 - 1}{10 \cdot 16} \cdot 48 \cdot 10^6}{16 \cdot 26} \text{ Hz} = \boxed{\frac{50.1\text{MHz}}{16 \cdot 26}}$$

- *Combining these two inequalities, we get the valid range of clock frequencies for the receiver:*
  - $46.25\text{MHz} \leq F_r \leq 50.1\text{MHz}$

Task 3 - Interrupt and Polling

Interrupts and Polling

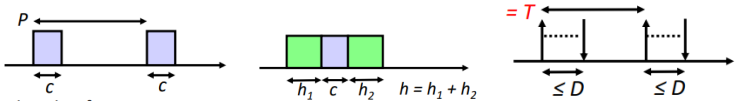
Repetition

- utilization u: average percentage the processor is busy
- computation c: processing time for handling the event
- overhead h: time overhead for handling the interrupt
- period P: polling period
- inter-arrival time T: minimal time between two events
- deadline D: maximal time between event arrival and finishing event processing with $D \leq T$.

Interrupts and Polling

Repetition

- Constraints for polling-based processing: $2c \leq P + c \leq D \leq T$
  - $P \geq C$ implies $P + C \geq 2C$
  - the maximal polling period is $P = D - c$, the minimal polling period is $P = c$
- Constraints for interrupt-based processing: $h + c \leq D \leq T$



Task 3 - Interrupts and Polling

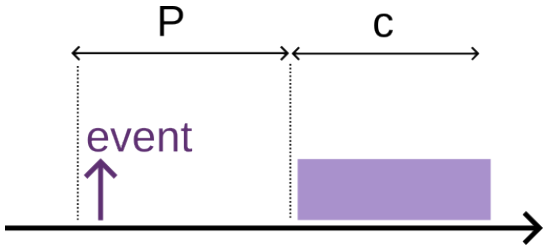
Task 3

Solution a:

- *Computation time c for the event:* $\frac{100}{48 \cdot 10^6 Hz} = 2.0833 \mu s$
- *The maximum response time in the worst case is* $P + c$*. This time should not exceed our deadline of* $10 \mu s$
- *Therefore, we have* $P + 2.0833 \mu s \leq 10 \mu s \iff P \leq 7.9167 \mu s$
- *Since our polling period may not be shorter than the computation time we obtain* $P \geq 2.0833 \mu s$*. Therefore, our range is* $P \in [2.0833 \mu s, 7.9167 \mu s]$
- *we're only interested in a lower bound for* P *not* $P + c$

Solution a:

Task 3



Task 3 - Interrupts and Polling

Task 3

Solution b:

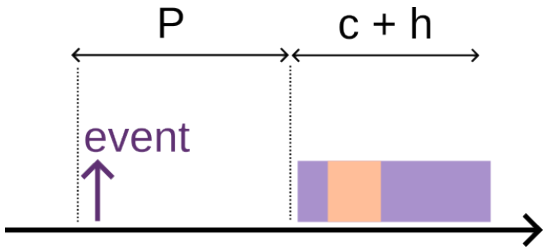
- *In this setting, the worst case occurs if an interrupt delays our event computation.*
- *Our maximum response time is $P + c + h$ where h is the overhead time required to process the interrupt.*
- *Because our event takes 100 cycles, and the minimum time between two interrupts is not less than 140 cycles, there can be at most one interrupt within the processing of our event.*
- *Thus, we might require 40 more cycles in the worst case, taking $\frac{40}{48 \cdot 10^6 Hz} = 0.8333 \mu s$.*

Solution b:

Task 3

Solution b:

- $2.0833\mu s + 0.8333\mu s + P \leq 10\mu s$
- $P \geq 2.08\mu s + 0.8333\mu s$
- *solving this equations for P we get* $P \in [2.9167\mu s, 7.0833\mu s]$



Task 3 - Interrupts and Polling

Task 3

Solution c:

- *Let E be the total cycles of computation within one polling period. Let k be the amount of interrupts that occured during the processing of our event.*
- *We have that $E = 100 + 40 \cdot k$*
- *1. To meet the deadline, we require $\frac{E}{48 \cdot 10^6 Hz} + P \leq 10\mu s$*
- *2. To finish the polling task before a new period $P \geq \frac{E}{48 \cdot 10^6 Hz}$*
- *Through this we can derive that $2 \cdot \frac{E}{48 \cdot 10^6 Hz} \leq 10\mu s \Rightarrow E \leq 240 Cycles$*

Task 3 - Interrupts and Polling

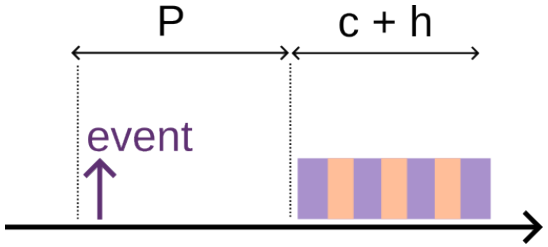
Task 3

Solution c:

- *Assuming the largest feasible value of E, we have* $k = \frac{240-100}{40} = 3.5$
- *This means at most 3 interrupts can be processed fully during the processing of one event. In total this takes* $\frac{100+40\cdot 3}{48\cdot 10^6} = 4.583\mu s$ *time.*
- *The minimum feasible time between two interrupts is* $T = 4.583\mu s/3 = 1.528\mu s$
- *The feasible range for the polling period is* $P \in [4.583\mu s, 5.147\mu s]$

Task 3 - Interrupts and Polling

Task 3



Bonus

Bonus I

Hexadecimal System

- Example: $\underline{beef}_{16} = 11 \cdot 16^3 + 14 \cdot 16^2 + 14 \cdot 16^1 + 15 \cdot 16^0$
  $= 11 \cdot 4096 + 14 \cdot 256 + 14 \cdot 16 + 15$
  $= 48879$

- all Bin and Hex assigned:

```
0    1    2    3    4    5    6    7    8    9
---- ---- ---- ---- ---- ---- ---- ---- ---- ----
0000 0001 0010 0011 0100 0101 0110 0111 1000 1001

A    B    C    D    E    F
---- ---- ---- ---- ---- ----
1010 1011 1100 1101 1110 1111
```

Bonus II

Hexadecimal System

- Hex ⇒ Bin:

```
   D    4    F    6    6    E
1101 0100 1111 0110 0110 1110
```

- Bin ⇒ Hex:

```
1101 0100 1111 0110 0110 1110
   D    4    F    6    6    E
```

Bonus III

Hexadecimal System

- Derivation:
  - $a4_{16} = 10 \cdot 16^1 + 4 \cdot 16^0$
    $= 10 \cdot (2^4)^1 + 4 \cdot (2^4)^0$
    $= 1010_2 \cdot 2^4 + 0100_2 \cdot 1$
    $= (1000_2 \cdot 2^4 + 10_2 \cdot 2^4) + (100_2 \cdot 2^0)$
    $= (1 \cdot 2^7 + 1 \cdot 2^5) + (1 \cdot 2^2)$
    $= 1010\_0100_2$
  - idea: shifting a number works in hexadecimal system $1a_{16} \cdot 10^2_{16} = 1a00$ decimal system $17 \cdot 10^2 = 1700$ and binary system $11_2 \cdot 10^2_2 = 1100_2$ quite similar.

Bonus IV

Hexadecimal System

- but beacuse $16 = 2^4$ the hexadecimal and binary system are particulary easy to convert into each other.

Bonus

Quiz question

- How many bits / digits will a hex number with 5 digits have in binary system?
  - ☐ 10
  - ☐ 15
  - ☐ 20
  - ☐ 16

Bonus

Quiz question

- How many bits / digits will a hex number with 5 digits have in binary system?
  - ☐ 10
  - ☐ 15
  - ■ 20
  - ☐ 16
- example: 0xa_aaaa = 0b1010_1010_1010_1010_1010
- the binary number has 4 times as many bits because $16 = 2^4$