Departament de Disseny i Programació de Sistemes Electrònics
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Embedded Systems
Final exam. June 12, 2013

Duration: 4 hours. Last revision day: June 26

1 Warming up questions

1. Define an embedded system.
2. Situate the course project DTMF in the previous definition of an embedded system.
3. What does the keyword CONST mean? This is one of the questions of a document entitled *Best Questions for Embedded Programmers*. Next you have an explanation given by Nigel Jones, an interviewer who tells us what should be answered and what not.

   As soon as the interviewee says 'const means constant,' I know I'm dealing with an amateur... suffice it to say that const means 'read-only'. Although this answer doesn't really do the subject justice, I'd accept it as a correct answer... If the candidate gets the answer correct, I'll ask him these supplemental questions: What do the following declarations mean?

   ```
   const int a;
   int const a;
   const int *a;
   int * const a;
   int const * a const;
   ```

   The first two mean the same thing, namely a is a const (read-only) integer. The third means a is a pointer to a const integer (that is, the integer isn't modifiable, but the pointer is). The fourth declares a to be a const pointer to an integer (that is, the integer pointed to by a is modifiable, but the pointer is not). The final declaration declares a to be a const pointer to a const integer (that is, neither the integer pointed to by a, nor the pointer itself may be modified). If the candidate correctly answers these questions, I'll be impressed. Incidentally, you might wonder why I put so much emphasis on const, since it is easy to write a correctly functioning program without ever using it. I have several reasons:

   a) The use of const conveys some very useful information to someone reading your code. In effect, declaring a parameter const tells the user about its intended usage. If you spend a lot of time cleaning up the mess left by other people, you'll quickly learn to appreciate this extra piece of information. (Of course, programmers who use const, rarely leave a mess for others to clean up.)

   b) const has the potential for generating tighter code by giving the optimizer some additional information

   c) Code that uses const liberally is inherently protected by the compiler against inadvertent coding constructs that result in parameters being changed that should not be. In short, they tend to have fewer bugs.

   In short, what does the CONST keyword mean?

   Is it useful when your code is used just for simulation (code size and code speed aren't important)?

2 C for embedded systems programming

1. You are using a 16-bit architecture microcontroller and you need a counter from 0 to 255. Which data type would you choose to declare this counter?

2. Do you know the size of a variable declared as *int*? What do you need to know? Is it related to TYPEDEF?

3. The keyword STATIC has three uses. Enumerate and explain in detail one of them.

4. Consider the following C code

```
uint8_t n=0;
int main(void){
    while(true){
    n++;
    }
return 0;
}
```

   and the following part of the assembly code inside the *main*, generated using the avr-gcc compiler with the optimizations options -Os and -mmcu=atmega328p.

```
.L2:
        rjmp .L2
```

   Discuss the result.

5. Consider the following C code

```
volatile uint8_t n=0;
int main(void){
    while(true){
    n++;
    }
return 0;
}
```

   and the following part of the assembly code inside the main, generated using the avr-gcc compiler with the optimizations options -Os and -mmcu=atmega328p.

```
.L2:
        lds r24,n (load direct from SRAM)
        subi r24,lo8(-(1)) (substract constant from register)
        sts n,r24 (store direct to SRAM)
        rjmp .L2
```

   Discuss the result.

6. Consider the following C code

```
volatile uint8_t n=0;
int main(void){
    while(true){
    n=n+2;
    }
return 0;
}
```

and write the part of the assembly code inside the main, generated using the avr-gcc compiler with the optimizations options -Os and -mmcu=atmega328p.

7. Consider the following C code

```
volatile uint8_t n=0;
int main(void){
    while(true){
    n++;
    n++;
    }
return 0;
}
```

and write the part of the assembly code inside the main generated using the avr-gcc compiler with the optimizations options -Os and -mmcu=atmega328p.

8. Look at the following rules for embedded designers and discuss their relation with code size and code speed.
   a) Use STATIC variables, whenever possible.
   b) Use CONST variables, whenever possible.
   c) Avoid VOLATILE variables, whenever possible.

3 C questions related to the course project DTMF

1. Look at the following precedence rules ordered from higher to lower priority.

```
Cast (type)a
Mult a*b a/b
Add a+b a-b
Shift a<<b b>>a
```

Regarding the Goertzel algorithm used in the course project DTMF and using the variables declared below (for the moment forget the promotion rules):

```
 int16_t s1=128,s2=4;sn=0;
 const int16_t a=8;
 uint8_t xn=8;
```

   a) Add all the parenthesis you need to make it clear what the following code does. Which is the value of sn?

```
 sn=xn+a*s1>>8-s2;
```

   b) Add all the parenthesis you need to make it clear what the following code does. Which is the value of sn?

```
 sn=a*s1>>8-s2+xn;
```

   c) Add all the parenthesis you need to make it clear what the following code does. Which is the value of sn?

```
 sn=xn-s2+a*s1>>8;
```

d) Write the right code, minimizing the number of parenthesis, that implements the filter used by the Goertzel algorithm: $sn = xn + ((a * s1)/256) - s2$. Which is the value of sn?

2. Considering promotion rules, explain in detail the size of each product and addition involved in the following code.

```
int32_t X=s1*s1+(int32_t)s1*s1+(int32_t)s2*s2-((int32_t)a*s1>>8)*s2;
```

3. Change the multiplication of the following code by additions and shifts.

```
uint16_t b,c;
...
c=b*a;
```

for the following values of a

a) `const uint16_t a=8;`

b) `const uint16_t a=254;`

c) `const uint16_t a=12;`

d) Which are the values of a that better improve execution time.

4 Raspberry Pi

1. Samples of 8 bit, sampled at 8 kHz, are send to the Raspberry Pi (RasPi) through an USB connection. What is the effective transmission rate of the USB connection to ensure that no data is lost? What transmission rate would you choose in your connection?

2. The following code is used to measure the maximum, mean and minimum time used by the RasPi to read a single byte (the buffer is always full). The results are 500 us, 400 us and 300 us.

```
import time
import numpy
import serial
ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1)

n=0
N=100
Dt=numpy.zeros((N,1))
L=1

while n<N:
  time.sleep(100e-3)
  t0=time.time()
  x=ser.read(L)
  t1=time.time()
  Dt[n,0]=t1-t0
  n=n+1

ser.close()

print "max is "+str(max(Dt))
print "mean is"+str(numpy.mean(Dt))
print "min is "+str(min(Dt))
```

   a) How many time, *reading time*, is needed to read a DTMF-window made up of 205 samples, one at a time, sampled at 8 kHz ?
   b) What percentage of the signal is lost?

3. Executing the previous code with $L = 205$, i.e. reading blocks of 205 samples at a time, approximately the same reading time results are obtained.
   a) Compute again the *reading time* that is needed to read a DTMF-window made up of 205 samples, 205 at a time .
   b) Are we losing samples?
   c) Now consider the *decision time* that is needed to make computations with these 205 samples, once they have been read, in order to decide the DTMF-tone received. How many time is left to make these computations before a new DTMF-window is read?

4. In the following consider that the *decision time* is 20 ms.
   a) What happens if occasionally the operative system of the RasPi makes us wait 100 ms to read a DTMF-window? Are we losing samples? Hint: condition your answer to the size of the USB-buffer.
   b) What do you prefer? A RasPi with a maximum reading time of 100 ms and a mean of 400 us or a bigger computer with a maximum of 800 ms and a mean of 50 us?
   c) How could you use the RasPi to detect DTMF-tones from two signals? Hint: Could you change the sampling frequency?

5. Comment on the differences between a real time system (RTS) and a non-real time system.

6. Which applications requires a RTS?

7. Could a real time operative system (RTOS) running on the RasPi solve the execution time problem found on the course project DTMF?

8. Could a code written in C instead of Python solve the execution time problem found on the course project DTMF?

5 FPGA and VHDL

1. Which are the main differences between a CPLD and an FPGA?

2. Do we need to know the hardware when using VHDL language?

3. Which of these statements can be used outside a *process*? IF, CASE, WHEN, WITH.

4. When compiling this code

```
process(b)
  begin
    if b='1' then
      test<='1';
    end if;
end process;
```

   we get this warning: *inferring latch(es) for signal or variable 'test'....*

   Why?

5. Comment on the differences between these processes. Do you know how are called some of them?

```
process(d,en)
  begin
    if en='1' then
      q<=d;
    end if;
end process;
```

```
process(d)
  begin
    if en='1' then
      q<=d;
    end if;
end process;
```

```
process(en)
  begin
    if en='1' then
      q<=d;
    end if;
end process;
```

```
process(en)
  begin
      q<=d;
end process;
```

6. A possible description of a D Flip-flop is:

```
process(clk,reset,set)
  begin
    if reset='0' then q<='0';
    elsif set='0' then q<='1';
    elsif rising_edge(clk) then q<=d;
    end if;
end process;
```

   a) Are the set and reset synchronous or asynchronous?
   b) If they are synchronous/asynchronous write the code to make them asynchronous/synchronous.
   c) What happens if the set and reset are low activated at the same time?

7. In the lab we have reduced the number of multipliers used to implement the Goertzel algorithm. Comment on when and how this is possible.

6 Choosing the right platform: FPGA, microcontroller or RasPi

1. Briefly discuss the advantages and disadvantages of each platform to carry out the course project DTMF.
2. Consider doubling the number of frequencies to be detected by the Goertzel algorithm. Comment on the effect it would have on each platform.
3. Which platform would you use to implement a battery of filters at high sampling rate?
4. Which platform would you use to communicate with several peripherals using I2C, ISP... read data from analog and digital inputs and write analog and digital data, all of this with low complexity decision algorithms?
5. Which of these platforms could be used in an application with real time requirements?
6. Comment on the use of an FPGA or a RasPi in your Systems Engineering course project (bed-head inclination).

7 Final question

If you know very well a topic that hasn't been asked yet this is your opportunity. You can briefly talk about one of these topics: CMOS, rate monotonic scheduling, debouncing, electrical noise and interferences, fixed-point arithmetic, DTMF codification, Goertzel algorithm..