Jonathan W. Valvano

(2) Question 1. life long learning

5. to **improve the understanding of technology**, its application, and consequences;
6. to **maintain and improve our technical competence**;

(10) Question 2. We studied three hardware software synchronization mechanisms

(2) Part a) Which method is least likely to cause a software crash (software stops running) when the hardware becomes faulty? **A)** Blind

(2) Part b) Which method is most efficient (creates the highest overall bandwidth) for handling many I/O channels with varying bandwidth? **C)** Interrupt

(2) Part c) Which method is most efficient (creates the highest overall bandwidth) for handling one I/O channel with varying bandwidth? **B)** Busy-wait

(2) Part d) Which method is best for implementing a situation with many I/O channels and one channel is very important? **C)** Interrupt

(2) Part e) Which method is best for implementing a real-time system with many I/O channels? **C)** Interrupt

(4) Question 3. Reduce power

(4) Question 4. A digital signal is interfaced to a microcontroller input. What happens to this digital interface when the effective capacitance to ground is increased?

E) the bandwidth decreases

(4) Question 5. What is the advantage of NRZ protocol over simple 3.3V/0V digital encoding?

D) both high and low require energy to communicate

(12) Problem 6. (4) Part a) Is there a critical section in the software system shown above?

D) yes, access to `y`

(2) Part b) How is the parameter **x** passed into the function? Not in general, but in this system?

A) Reg R0

(2) Part c) Where is the variable **out** allocated in the main program?

F) The compiler optimized this so much the parameter was removed

or **A)** Reg R0

(2) Part d) What does the `const` qualifier in the function `LowPassFilter()` mean?

D) the value is fixed and can not be changed by the function

(2) Part e) What does the `static` qualifier in the function `LowPassFilter()` mean?

F) stored in permanent RAM

(2) Part f) How does the return from interrupt instruction `POP {pc}` change context?

D) pops 0xFFFFFFF9, then pops 8 more registers

(4) Question 7. What is the fundamental justification for the Shannon Channel Capacity Theorem?

B) When defining an encoding scheme, we wish to separate the codes further apart than the noise level. The signal to noise ratio gives you the number of bits per code. The SCCT states that multiplying this by the number by the number of codes per second yields maximum possible bandwidth.

(8) Question 8. Multiply by 1000, divide by 1000

$y = (1428 \cdot x_0 + 1071 \cdot x_1 + 750 \cdot y)/1000$

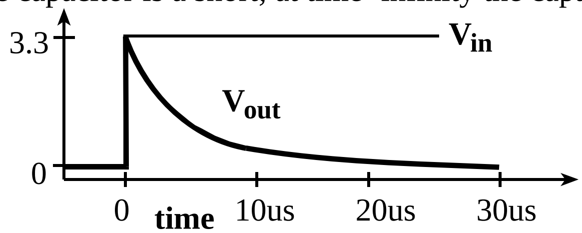
or $y = (4 \cdot x_0 + 3 \cdot x_1 + 21 \cdot y)/28$

(4) Question 9. Producer rate 20,000 bytes/sec. Consumer rate 10,000 bytes/sec.

D) The system does not work, but could be corrected by increasing baud rate to 200,000 bps.

(2) Question 10. Git is a version control system allowing many programmers to work simultaneously on one project

(6) Question 11. A time=0, the capacitor is a short; at time=infinity the capacitor is an open



(4) Question 12. State the ADC parameter determined by each.

Part a) Resolution.

Part b) Precision.

Part c) Repeatability. Noise level.

Part d) Linearity.

(6) Question 13. An overflow error occurred if we add to positive numbers and get a negative sum. Similarly, an overflow error occurred if we add to negative numbers and get a positive sum. All other combinations are not error.

x	y	x+y	overflow
positive	positive	positive	ok
positive	positive	negative	error
positive	negative	positive	ok
positive	negative	negative	ok
negative	positive	positive	ok
negative	positive	negative	ok
negative	negative	positive	error
negative	negative	negative	ok

```
unsigned long Error=0;
long DigitalFilter(long x){ static long y; long sum=x+y;
  if(((x>0)&&(y>0)&&(sum<0)||((x<0)&&(y<0)&&(sum>0))) Error++;
  y = (x+y)/2;
  return y;
}
```

(10) Question 14. Since we do not wish to waste power, R2 and R3 are 0. The collector current is 200mA, so the base current needs to be 200mA/50 = 4mA. R1 controls base current. If you use V_{OH}=3.3V, then

R1 = (3.3-0.9)V / 4mA = 2.4V/4mA = 600 Ω.

If you use V_{OH}=2.4V, then

R1 = (2.4-0.9)V / 4mA = 1.5V/4mA = 375 Ω.

Any value between 100Ω and 600Ω is ok.

(10) Question 15. First, we show an indexed solution,

```
struct State {
  unsigned char Out;
  unsigned char In[2];
  unsigned char Next[2];};
typedef const struct State STyp;
#define S1  0
#define S2  1
#define S2  2
STyp FSM[3]={
 { 25,{ 21, 99},{S2,S3}},   // S1
 {100,{120,  3},{S1,S3}},   // S2
 { 50,{  0,  5},{S1,S2}}};  // S3
void FSM(void){ unsigned char n; // state number
  unsigned char input;
  n = S1;
  while(1){
    GPIO_PORTD_DATA_R = FSM[n].Out;
    input = GPIO_PORTF_DATA_R;
    if(input==FSM[n].In[0]) n = FSM[n].Next[0];
    if(input==FSM[n].In[1]) n = FSM[n].Next[1];
  }
}
```

Next, we show a pointer solution (this will run a little faster, but needs a little more ROM memory to store the structure)

```
struct State {
  unsigned char Out;
  unsigned char In[2];
  const struct State *Next[2];};
typedef const struct State STyp;
#define S1 &FSM[0]
#define S2 &FSM[1]
#define S3 &FSM[2]
STyp FSM[3]={
 { 25,{ 21, 99},{S2,S3}},   // S1
 {100,{120,  3},{S1,S3}},   // S2
 { 50,{  0,  5},{S1,S2}}};  // S3
void FSM(void){ STyp *pt;   // state pointer
```

```
  unsigned char input;
  pt = goN;
  while(1){
    GPIO_PORTD_DATA_R = pt->Out;
    input = GPIO_PORTF_DATA_R;
    if(input == pt->In[0]) pt = pt->Next[0];
    if(input == pt->In[1]) pt = pt->Next[1];
  }
}
```

Third, we show a pointer solution with the fastest execution. This will run much much faster, but needs a lot more ROM memory to store the structure. Of course most microcontrollers have lots of ROM, so this approach is appropriate. It is difficult to type in, but check out how fast it runs. It still has one-to-one mapping, although it is more likely to have a typing error in the FSM than the above solutions.

```
struct State {
  unsigned char Out;
  const struct State *Next[256];};
typedef const struct State STyp;
#define S1 &FSM[0]
#define S2 &FSM[1]
#define S3 &FSM[2]
STyp FSM[3]={
 { 25,                            // S1
{S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S2,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,
 S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,
 S1,S1,S1,S1,S1,S1,S1,S3,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,
 S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,
 S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,
 S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1,S1},
 {100,                           // S2
{S2,S2,S2,S3,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,
 S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,
 S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S1,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,
 S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,
 S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,
 S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2,S2},
 { 50,                           // S3
{S1,S3,S3,S3,S3,S2,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,
 S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,
 S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,
 S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,
 S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,
 S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3,S3}
};
void FSM(void){ STyp *pt;   // state pointer
  unsigned char input;
  pt = goN;
  while(1){
    GPIO_PORTD_DATA_R = pt->Out;
    pt = pt->Next[GPIO_PORTF_DATA_R];
  }
}
```

(10) Question 16. You will design the analog hardware for a data acquisition system to measure pressure.

Part a) Notice that R1+R3=R2+R4=2000 always, 990/2000*2.5=1.2375, 1010/2000*2.5=1.2625

Pressure (mmHg)	R1 (Ω)	R2 (Ω)	R3 (Ω)	R4 (Ω)	V2 (V)	V1 (V)	V2-V1 (V)	V3 (V)	ADC result
0	1000	1000	1000	1000	1.25	1.25	0	0	0
100	995	1005	1005	995	1.25625	1.24375	0.0125	1.5	512
200	990	1010	1010	990	1.2625	1.2375	0.025	3	1023

Part b) Gain needed is 3/0.025=120. No offset is needed because 0 maps to 0, so LM4041 is not needed. Use an instrumentation amp to get differential gain and good CMRR is required. INA122/AD627 gain is 5+200k/**Rg**, for this amp **Rg** = 200k/115= 1.74k.