EE 3220 – System-on-Chip Design

Q1:

A. What are the inputs?

Water level Sensor, Touch key, Temp Sensor, Flow Sensor

B. What are the outputs?

Pump motor, Grinding motor, LCD, GPIO

C. Name at least two safety features to include, and specify what hardware and/or software is needed for each.

Alarm when run out of coffee beans

Hardware: Speaker, Coffee Bean Sensor,

Software: Function to alert suer when coffee bean level is low

Alarm when there is no water

Hardware: Speaker, Water level Sensor

Software: Function to alert suer there is no water

D. Describe a useful feature which could you add in software without requiring additional hardware.

Turn off when time out.

E. Describe a useful feature which could you add in software if the controller could keep track a user profile setting and add a new I/O component.

Control how much milk to add in the coffee

F. Describe a useful feature which could you add in software if the coffee machine included an internet connection.

Control the coffee machine with an application within the user's smart phone

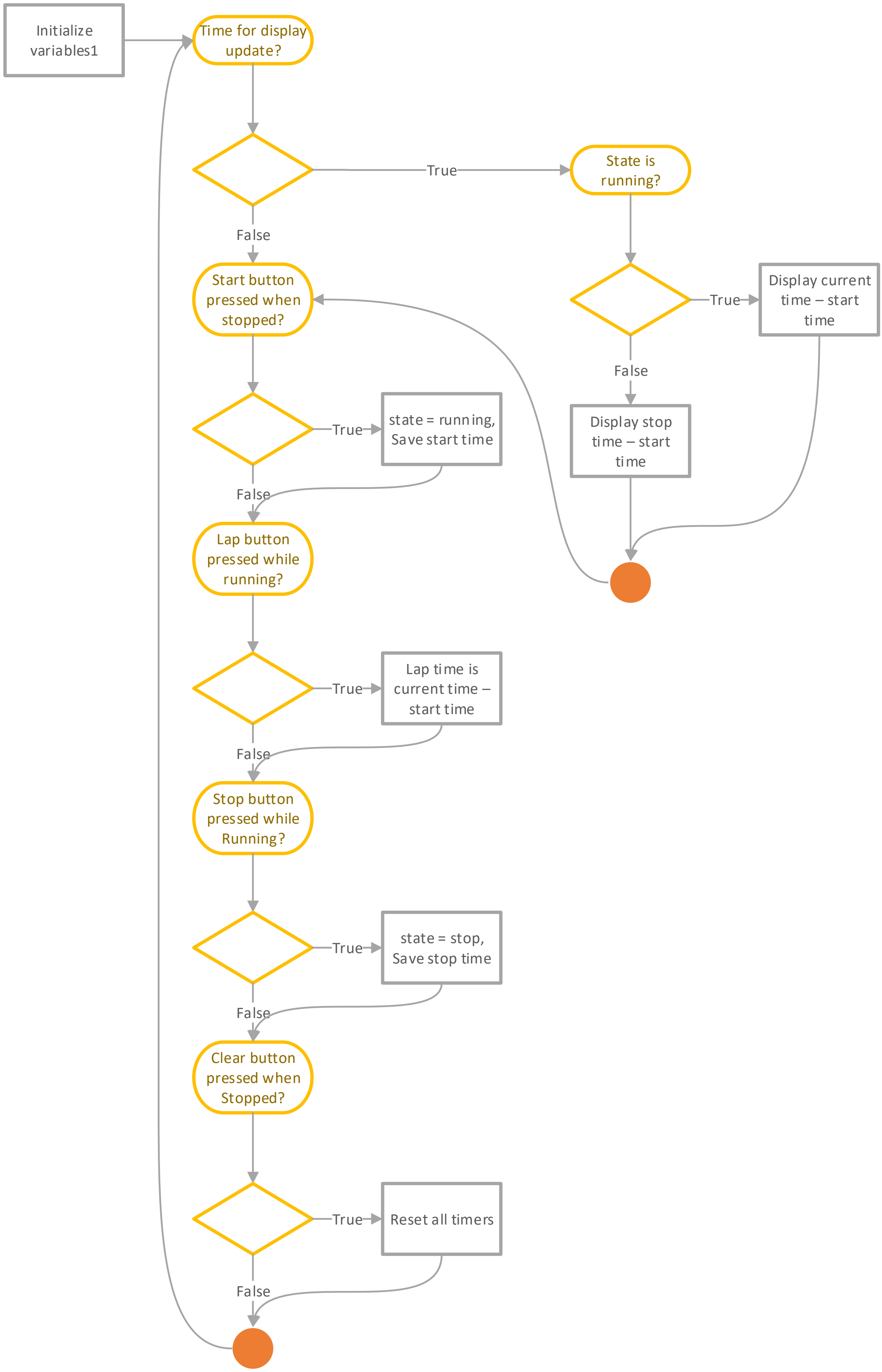
Q2:

```
start_ISR()
{
    if (state == 0)
        state = 1;
}

stop_ISR()
{
    if (state == 1)
        state = 0;
}

clear_ISR()
{
    if (state == 0)
        elapsed_time = lap_time = 0;
}

lap_ISR()
{
    if (state != 0)
        lap_time = elapsed_time;
}

timer_ISR()
{
    if (state != 0)
        elapsed_time++;

    if (state == 1 && display_delay > 0)
        display_delay--;

    if (display_delay == 0)
    {
        state = 2;
        display_delay = 100;
        display(elapsed_time);
    }
}

main()
{
    state = 0; // state 0: stopped, state 1: running, state 2: recording
    display_delay = 100;
    elapsed_time = lap_time = 0;

    while (true)
        ; // main loop
}
```

Q3:

```
main()
{
    // state 0: stopped, state 1: running
    next_display_update = state = 0;
    start_time = stop_time = lap_time = 0;                          // time variable
    start_pressed = stop_pressed = lap_pressed = clear_pressed = 0; // button

    display(elapsed_time_register);

    while (true)
    {
        if (elapsed_time_register % 100 == 0)
            next_display_update = 1;

        if (next_display_update)
        {
            if (state = 1)
                display(elapsed_time_register - start_time);
            if (state = 0)
                display(stop_time - start_time);
            next_display_update = 0;
        }

        if (state == 0 && start_pressed)
        {
            state = 1;
            start_pressed = 0;
            start_time = elapsed_time_register;
            // run_state();
        }

        if (state == 1 && lap_pressed)
        {
            lap_time = elapsed_time_register - start_time;
            lap_pressed = 0;
        }

        if (state == 1 && stop_pressed)
        {
            state = stop_pressed = 0;
            stop_time = elapsed_time_register;
        }

        if (state == 0 && clear_pressed)
        {
            start_time = stop_time;
            clear_pressed = 0;
        }
    }
}
```

Q4.


Initialize variables1
Time for display update?
True
False
State is running?
Start button pressed when stopped?
True
Display current time – start time
False
Display stop time – start time
True
state = running, Save start time
False
Lap button pressed while running?
True
Lap time is current time – start time
False
Stop button pressed while Running?
True
state = stop, Save stop time
False
Clear button pressed when Stopped?
True
Reset all timers
False

Q5:

What is the ARM Architecture used in this SoC?

Armv9

What kind of new instructions are introduced in this Architecture?

Security, AI, and improved vector (SVE, SVE2) and DSP capabilities

Please introduce the basic components used in this SoC.

Qualcomm Adreno GPU

Qualcomm Kryo CPU

Snapdragon X65 5G Modem-RF System

Qualcomm FastConnect 6900 System

Qualcomm Spectra Image Signal Processor

Qualcomm Aqstic audio codec

Qualcomm Aqstic smart speaker amplifier

This SoC can achieve high performance and power saving. In your own words, how this SoC is able to achieve these outstanding features as compared to State-of-the-art products?

This SoC is built based on Arm v9 architecture, which is a RISC based processor architecture. This architecture is more power efficiency comparing to other processor architectures. This SoC also has many components to accelerate specific features like having the Snapdragon X65 5G Modem-RF System intergrade in order to achieve high performance on 5G networking.

Q6:

$$die\ yield = \left(1 + \frac{defects\ pre\ unit\ area \times die\ area}{\alpha}\right)^{-\alpha}$$
$$= \left(1 + \frac{1 \times 2.5}{3}\right)^{-3}$$
$$= 0.1623$$

Q7:

$$area = 100 \times \left(\frac{1}{2}\right)^4$$
$$= 6.25mm^2$$
$$precentage = \frac{6.25}{100}$$
$$= 6.25\%$$

Q8:

A. Static, non-preemptive scheduler:

Best Case: E start first, then ABCD

Worst Case: ABCD start first, then E

Minimum response times: 5

Maximum response times: 15

B. Dynamic, non-preemptive scheduler: EABCD

Best Case: E start first, then ABCD

Worst Case: C start first, then E

Minimum response times: 5

Maximum response times: 9

C. Dynamic, preemptive scheduler: EABCD

Best Case: E start first, then ABCD

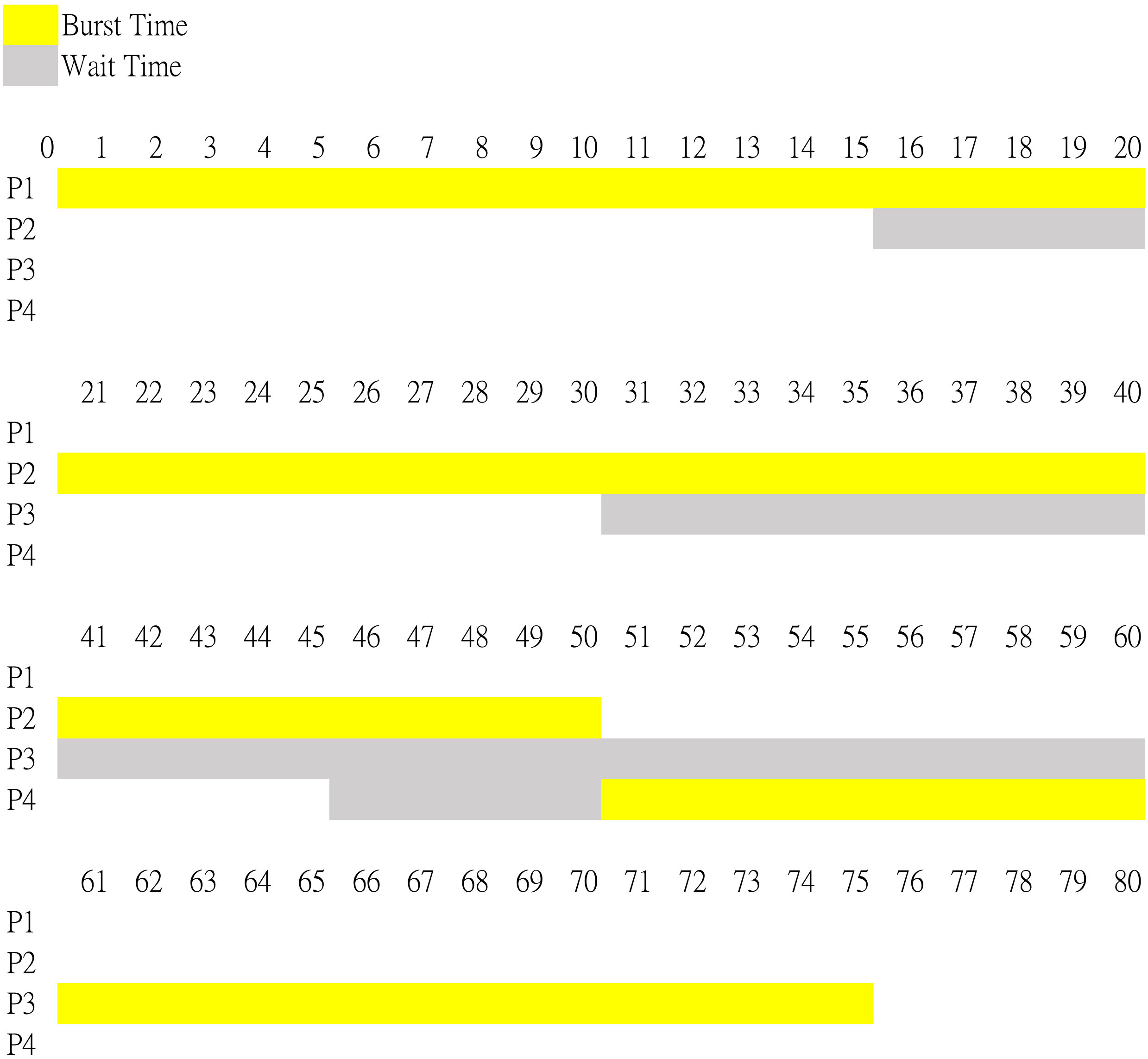
Worst Case: C start first, then it is preempted by E

Minimum response times: 5

Maximum response times: $5 + n$ which $n < 4$

Q9:

Process	Arrival time	**Burst time**	Service Time	**Wait time**
P1	0	20	0	0
P2	15	30	20	5
P3	30	15	60	30
P4	45	10	50	5
Average				10



Q10:

Process	Arrival time	**Burst time**	Service Time	**Wait time**
P1	0	9	0	12
P2	1	4	3	11
P3	2	5	6	11
P4	4	3	9	5
Average				9.75