Introduction to Lab
SIRD Model

José Nelson Amaral

The SIRD Model

A compartmental model used to simulate the spread of infectious diseases.

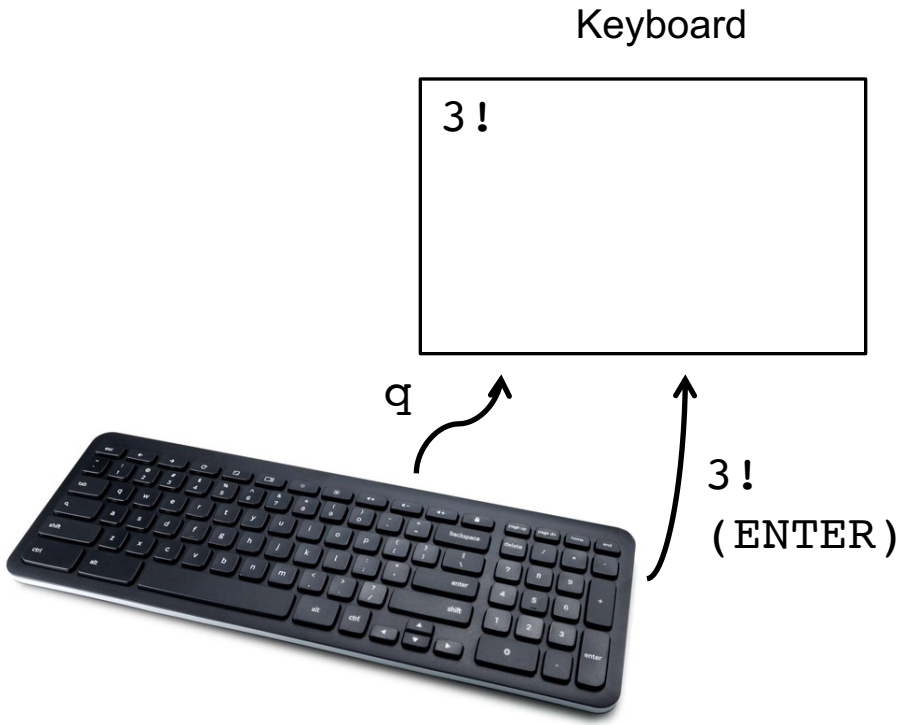
At any given moment, each member of the population belongs to one of the following compartments:

- **Susceptible:** at risk of contracting the disease
- **Infectious:** currently suffering from the disease
- **Recovered:** recovered from the disease
- **Deceased:** killed by the disease

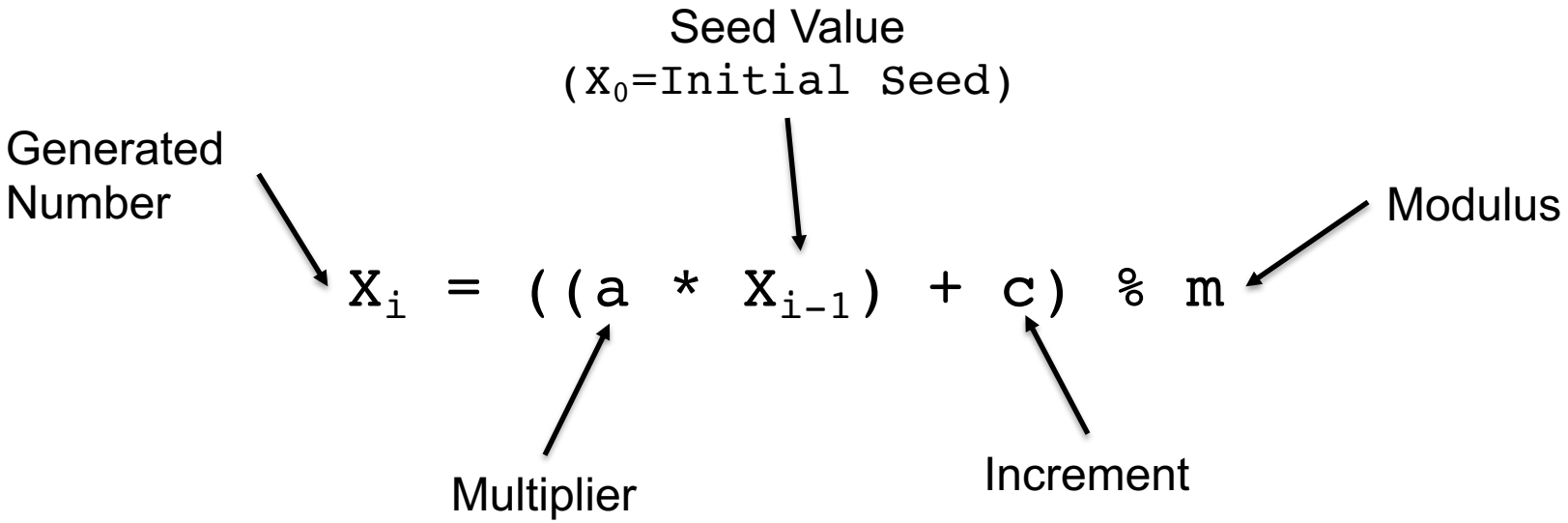
An Infectious Disease Simulator

Display

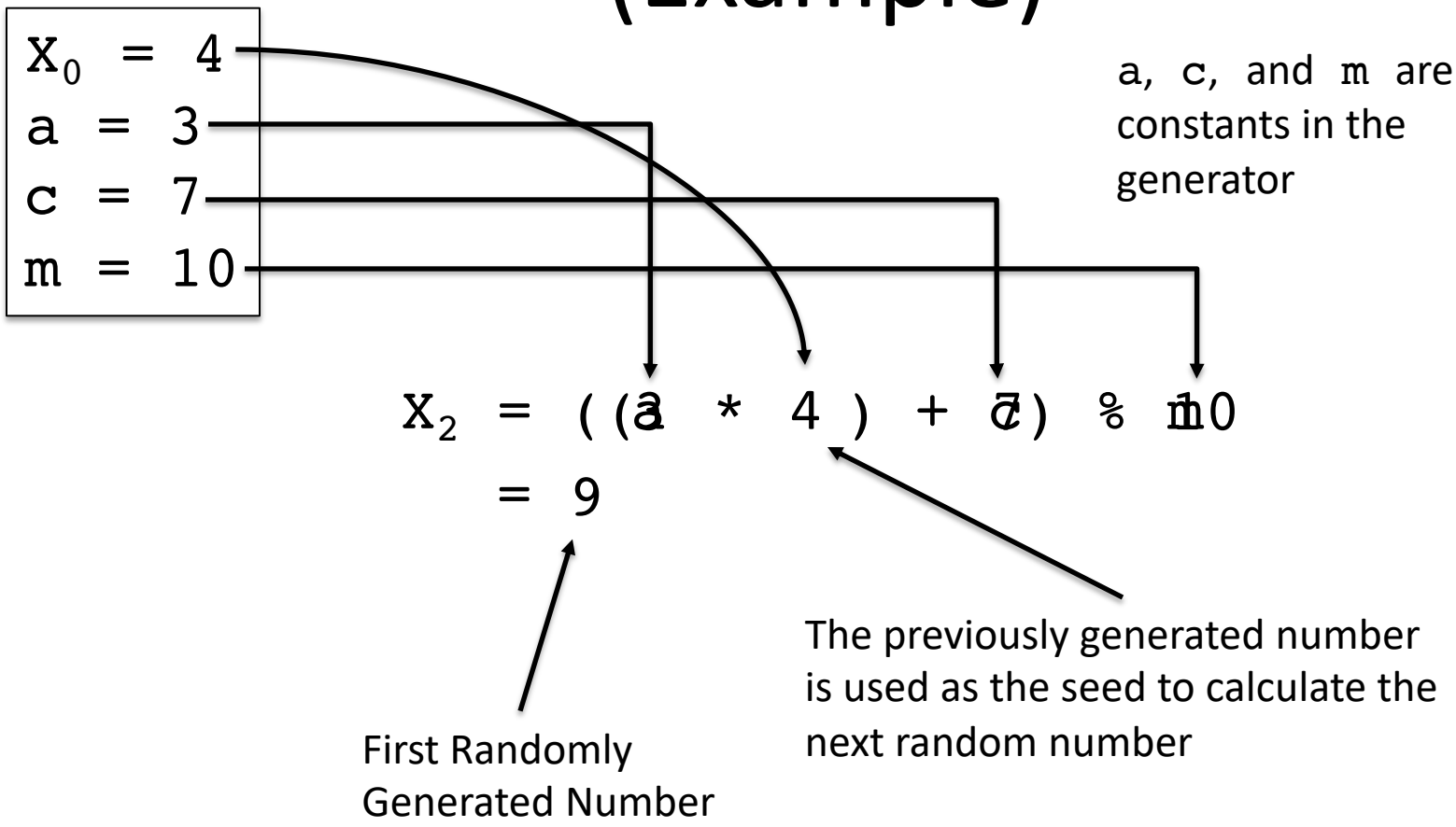
```
Person 0 - S
Person 1 – S
Person 2 – S
Person 3 – I:02
Person 4 – S
Person 5 – S
Person 6 – S
Person 7 – S
Person 8 - s
Person 9 - S
```



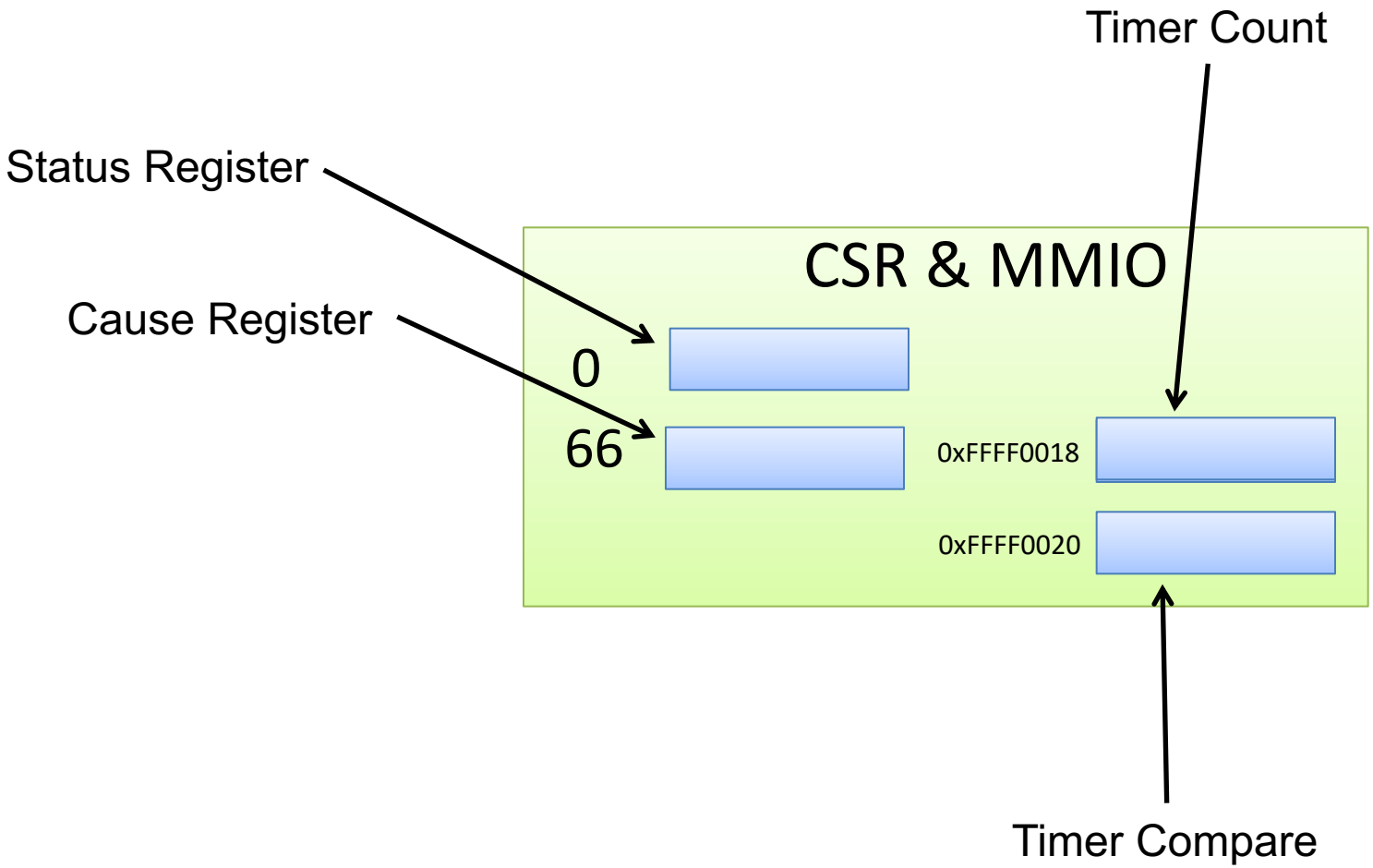
Linear Congruential Generator

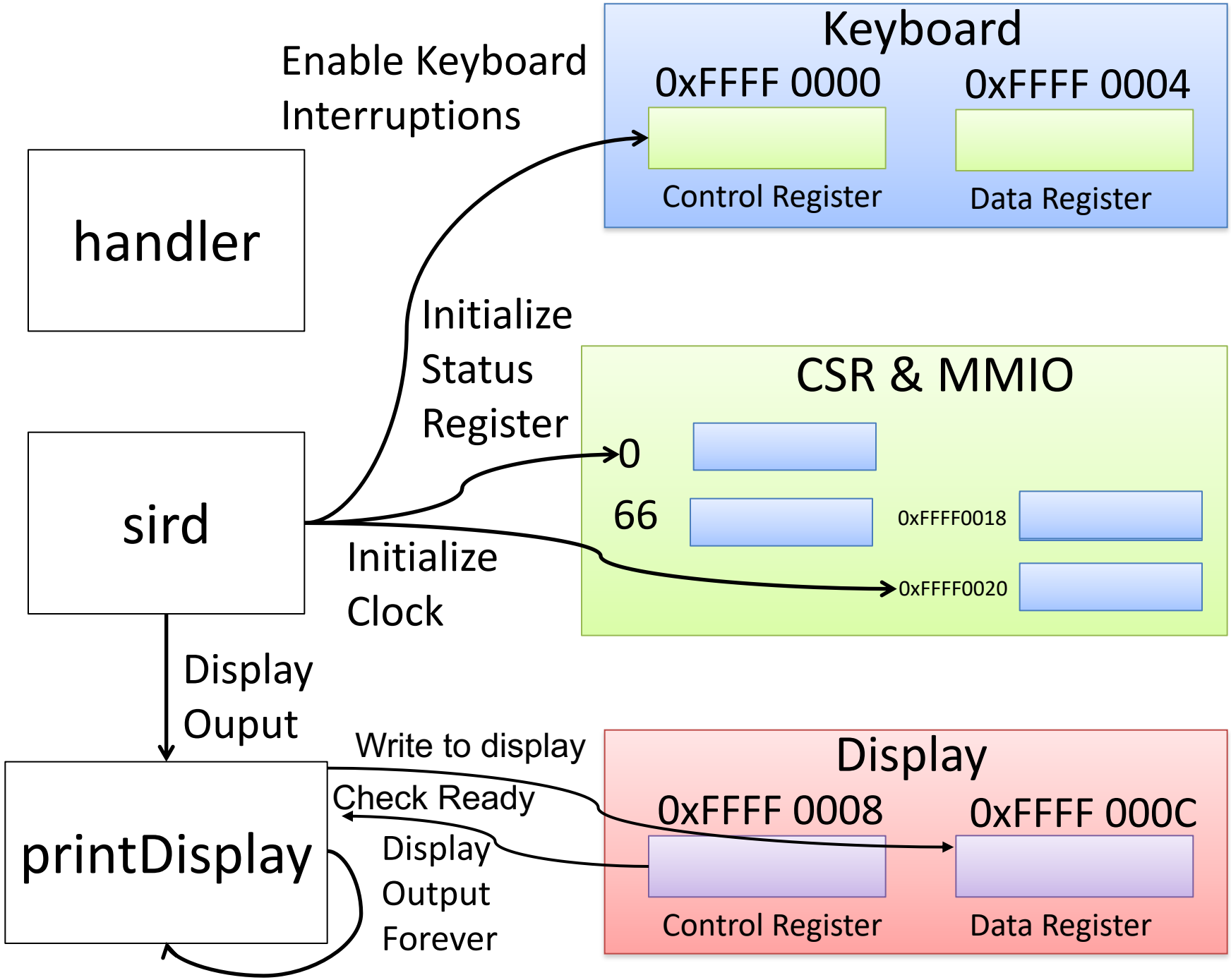


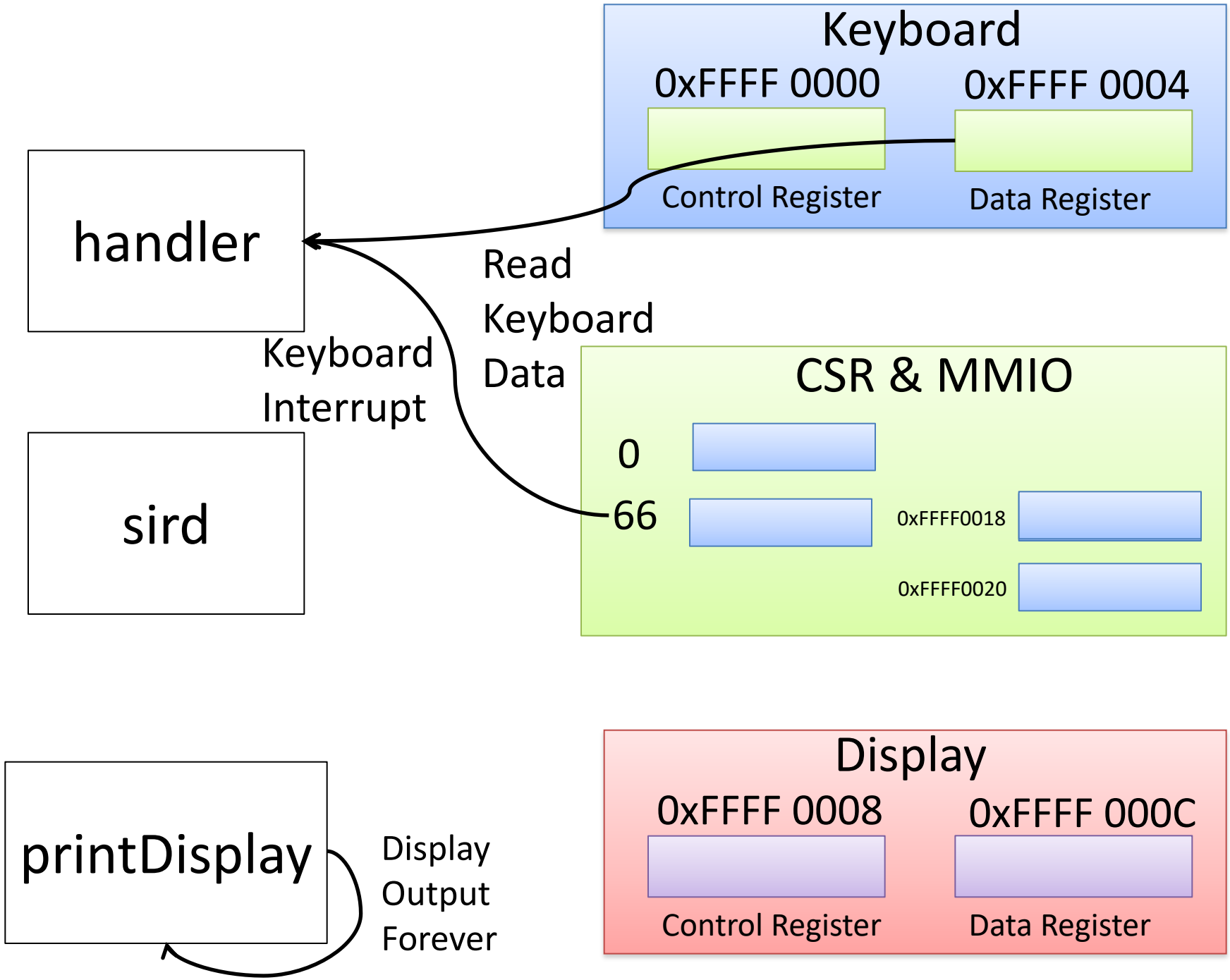
Linear Congruential Generator (Example)

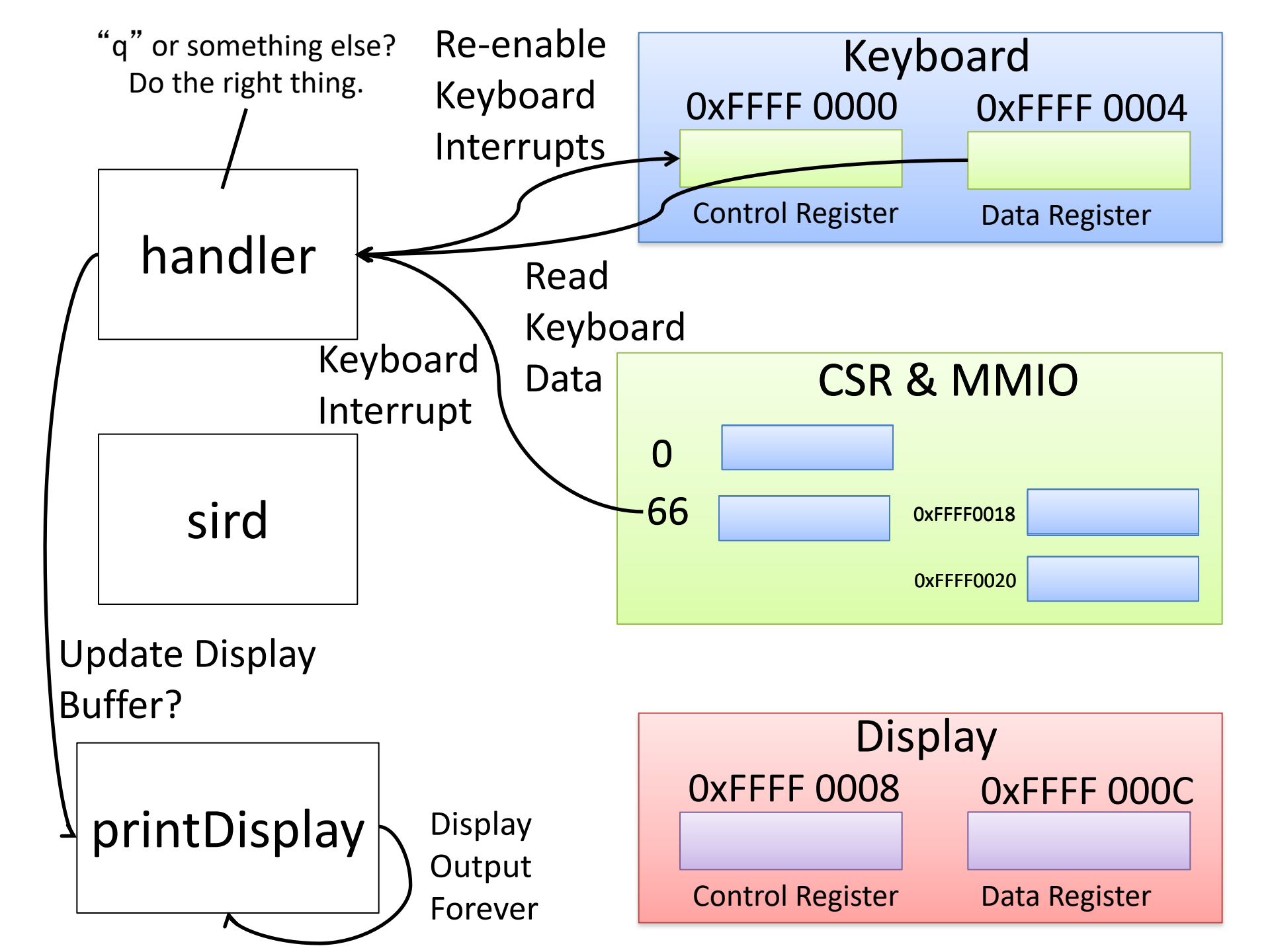


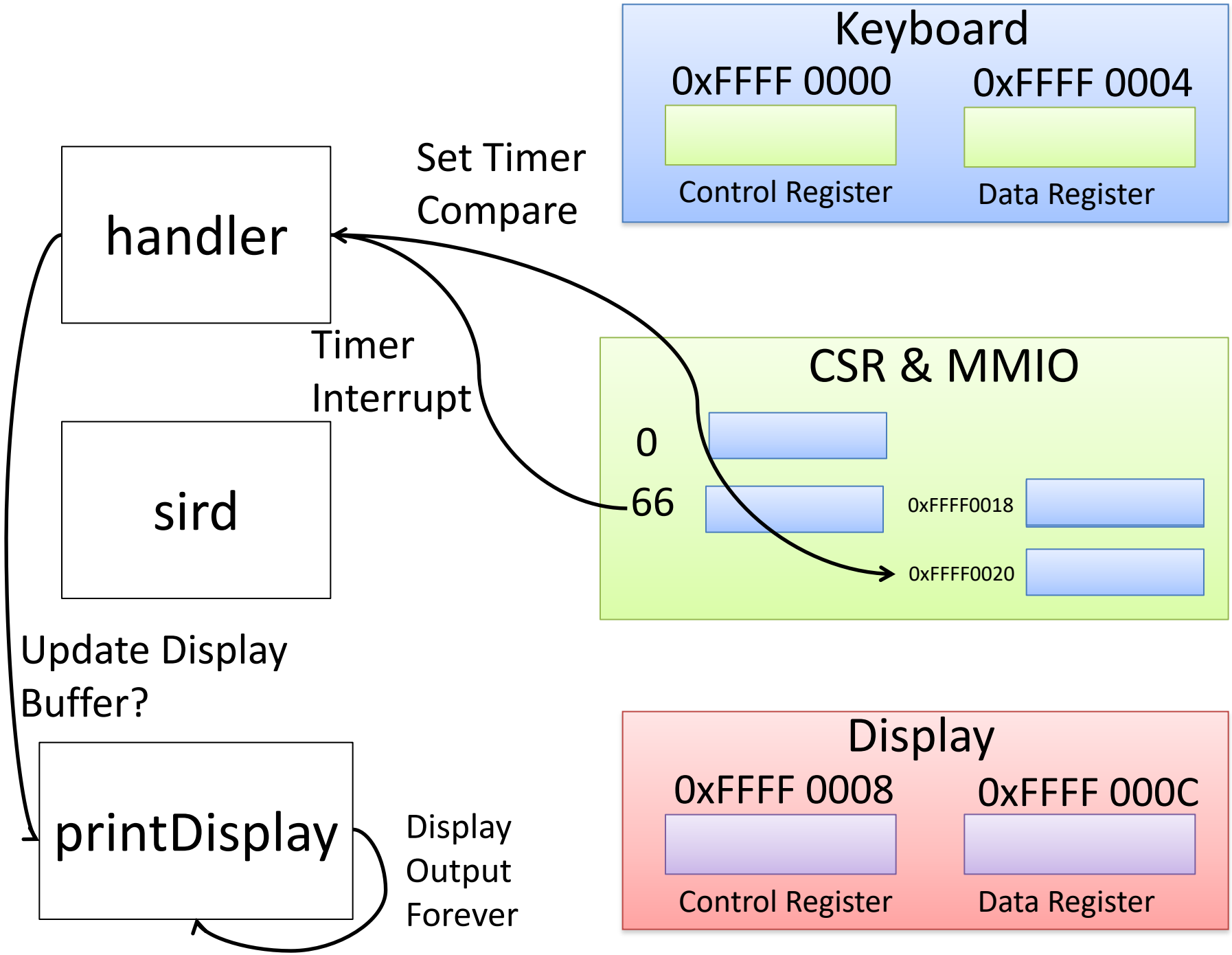
CSR & MMIO


Timer Count
Status Register
Cause Register
CSR & MMIO
0
66
0xFFFF0018
0xFFFF0020
Timer Compare


Keyboard
0xFFFF 0000
0xFFFF 0004
Control Register
Data Register
Enable Keyboard Interruptions
handler
Initialize Status Register
CSR & MMIO
0
66
0xFFFF0018
0xFFFF0020
sird
Initialize Clock
Display Ouput
Write to display
Check Ready
Display
0xFFFF 0008
0xFFFF 000C
Control Register
Data Register
printDisplay
Display Output Forever


Keyboard
0xFFFF 0000
0xFFFF 0004
Control Register
Data Register
handler
Read
Keyboard
Data
Keyboard
Interrupt
sird
CSR & MMIO
0
66
0xFFFF0018
0xFFFF0020
printDisplay
Display
Output
Forever
Display
0xFFFF 0008
0xFFFF 000C
Control Register
Data Register


"q" or something else?
Do the right thing.
Re-enable
Keyboard
Interrupts
Keyboard
0xFFFF 0000
0xFFFF 0004
Control Register
Data Register
handler
Read
Keyboard
Data
Keyboard
Interrupt
CSR & MMIO
0
66
0xFFFF0018
0xFFFF0020
sird
Update Display
Buffer?
printDisplay
Display
Output
Forever
Display
0xFFFF 0008
0xFFFF 000C
Control Register
Data Register


Keyboard
0xFFFF 0000
0xFFFF 0004
Control Register
Data Register
handler
Set Timer
Compare
Timer
Interrupt
sird
CSR & MMIO
0
66
0xFFFF0018
0xFFFF0020
Update Display
Buffer?
printDisplay
Display
Output
Forever
Display
0xFFFF 0008
0xFFFF 000C
Control Register
Data Register

```
    p = address of first character
FOREVER:
```

With this solution the timer will be blinking

Saving Registers

- An interrupt handler must save all the registers it uses so that the program can resume execution when the handler returns.
- The label `iTrapData` designates a section of memory allocated for saving registers in the handler.
- Outside of the handler, `uscratch` (CSR #64) should contain the address of the `iTrapData` section.
- Use the `cssrw` instruction to swap a register with the `uscratch` and save all the required registers.

Saving Registers (Sample Code)

```
handler:
```