The factorial file controls the stack pointer

The timer file is a timer for delay

The kernel is the main file

Register 1 and the input is #4

1. kernal

;Calculate

mov r1,#4 ;input

mov sp,$1000 ;make room on the stack

mov r0,r1

bl FACTORIAL

mov r7,r0 ;store answer

BASE = $3F000000 ;RP2 and RP3 ;GPIO\_SETUP

GPIO\_OFFSET = $200000

mov r0,BASE

bl SETUP\_LED

FLASH:

;param r0=BASE

;param r1 = number of flashes

orr r0,GPIO\_OFFSET

mov r7,r1

loop$:

mov r1,#1

lsl r1,#18

str r1,[r0,#28] ;turn LED on

mov r2,$0F0000 ;not using r2 for anything else so no need to push/pop

push {lr}

bl TIMER

pop {lr}

mov r1,#1

lsl r1,#18

str r1,[r0,#40] ;turn LED off

mov r2,$0F0000

push {lr}

bl TIMER

pop {lr}

sub r7,#1

cmp r7,#0

bne loop$ ;end of outer loop. Runs r7 times

bx lr

loop$:

mov r1,#1

lsl r1,#18

str r1,[r0,#28] ;turn LED on

mov r2,$0F0000 ;not using r2 for anything else so no need to push/pop

bl TIMER

mov r1,#1

lsl r1,#18

str r1,[r0,#40] ;turn LED off

mov r2,$0F0000

bl TIMER

sub r7,#1

cmp r7,#0

bne loop$ ;end of outer loop. Runs r7 times

wait:

b wait

include "TIMER.asm"

include "factorialj.asm"

SETUP\_LED:

;param r0=BASE

orr r0,GPIO\_OFFSET

mov r1,#1

lsl r1,#24

str r1,[r0,#4] ;set GPIO18 to output

bx lr

2.1 Kernal

;kernel7.asm

;Calculate

mov r1,#4 ;input

mov sp,$1000 ;make room on the stack

mov r0,r1

bl FACTORIAL

mov r7,r0 ;store answer

BASE = $3F000000 ;RP2 ;GPIO\_SETUP

mov r0,BASE

bl SETUP\_LED

mov r0,BASE

bl SETUP\_LED

mov r0,BASE

mov r1,r7

bl FLASH

wait:

b wait

include "TIMER.asm"

include "factorialj.asm"

include "GPIO.asm"

2.2 gpio

;Calculate

mov r1,#4 ;input

mov sp,$1000 ;make room on the stack

mov r0,r1

bl FACTORIAL

mov r7,r0 ;store answer

BASE = $3F000000 ;RP2 and RP3 ;GPIO\_SETUP

GPIO\_OFFSET = $200000

mov r0,BASE

orr r0,GPIO\_OFFSET

mov r1,#1

lsl r1,#24

str r1,[r0,#4] ;set GPIO18 to output

loop$:

mov r1,#1

lsl r1,#18

str r1,[r0,#28] ;turn LED on

mov r2,$0F0000 ;not using r2 for anything else so no need to push/pop

bl TIMER

mov r1,#1

lsl r1,#18

str r1,[r0,#40] ;turn LED off

mov r2,$0F0000

bl TIMER

sub r7,#1

cmp r7,#0

bne loop$ ;end of outer loop. Runs r7 times

wait:

b wait

include "TIMER.asm"

include "factorialj.asm"

push {lr}

bl TIMER

pop {lr}

sub r7,r7,#1

cmp r7,#0

bgt loop$ ;end of outer loop. Runs r7 times

bx lr