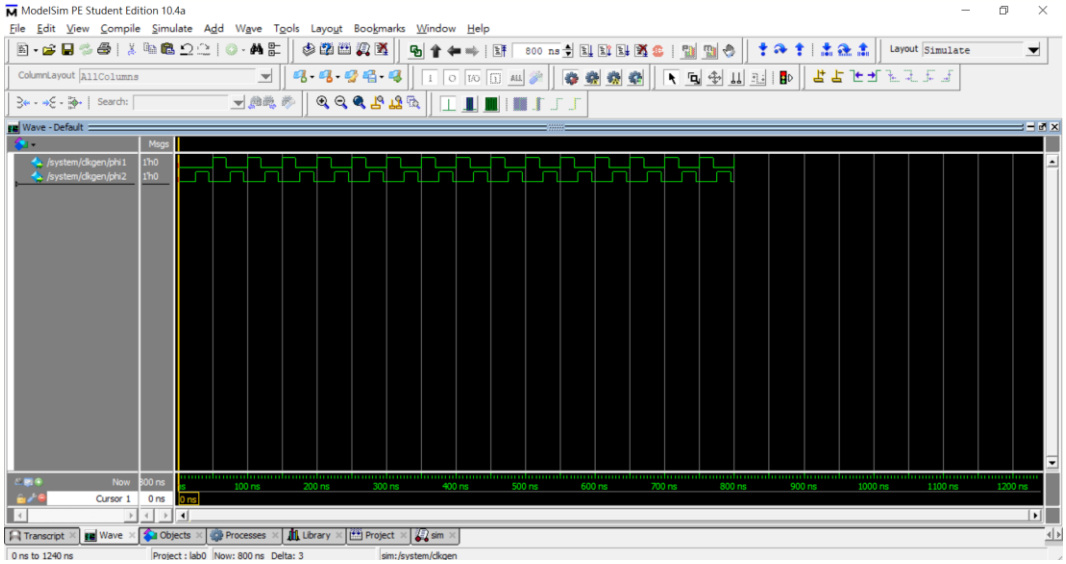
Part 1:




Part 2:

```
multiplication(int, int):              # @multiplication(int, int)
        addi    sp, sp, -32            sp=sp-32
        sd      ra, 24(sp)             Memory[sp+24]=ra
        sd      s0, 16(sp)             Memory[sp+16]=s0
        addi    s0, sp, 32             s0=sp+32
        add     a2, zero, a1           a2=zero+a1
        add     a3, zero, a0           a3=zero+a0
        sw      a0, -20(s0)            Memory[s0-20]=a0
        sw      a1, -24(s0)            Memory[s0-24]=a1
        lw      a0, -20(s0)            a0=Memory[s0-20]
        lw      a1, -24(s0)            a1=Memory[s0-24]
        mulw    a0, a0, a1             a0=a0*a1
        ld      s0, 16(sp)             s0=Memory[sp+16]
        ld      ra, 24(sp)             ra=Memory[sp+24]
        addi    sp, sp, 32             sp=sp+32
        ret                            return
main:                                  # @main
        addi    sp, sp, -32            sp=sp-32
        sd      ra, 24(sp)             Memory[sp+24]=ra
        sd      s0, 16(sp)             Memory[sp+16]=s0
        addi    s0, sp, 32             s0=sp+32
        addi    a0, zero, 2            a0=zero+2
```

```
sw      a0, -20(s0)                 Memory[s0-20]=a0
addi    a0, zero, 3                 a0=zero+3
sw      a0, -24(s0)                 Memory[s0-24]=a0
lw      a0, -20(s0)                 a0=Memory[s0-20]
lw      a1, -24(s0)                 a1=Memory[s0-24]
call    multiplication(int, int)    call    multiplication(int, int)
sw      a0, -28(s0)                 Memory[s0-28]=a0
mv      a0, zero                    a0=zero
ld      s0, 16(sp)                  s0=Memory[sp+16]
ld      ra, 24(sp)                  ra=Memory[sp+24]
addi    sp, sp, 32                  sp=sp+32
ret                                 return
```