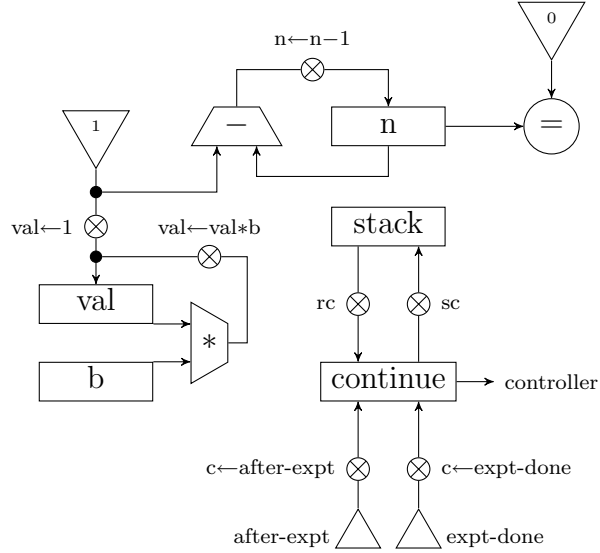a. For the recursive exponentiation, we use the following data paths diagram



```
( controller
    ( assign continue ( label expt−done ))
  expt−loop
    ( test ( op =) ( reg n) ( const 0))
    ( branch ( label base−case ))
    ( save continue )
    ( assign continue ( label after−expt ))
    ( assign n ( op −) ( reg n) ( const 1))
    ( goto ( label expt−loop ))
  after−expt
    ( restore continue )
    ( assign val ( op *) ( reg val) ( reg b))
    ( goto ( reg continue ))
  base−case
    ( assign val ( const 1))
    ( goto ( reg continue ))
  expt−done )
```

b. The iterative exponentiation doesn't need a stack since the recursive call is in tail call position.
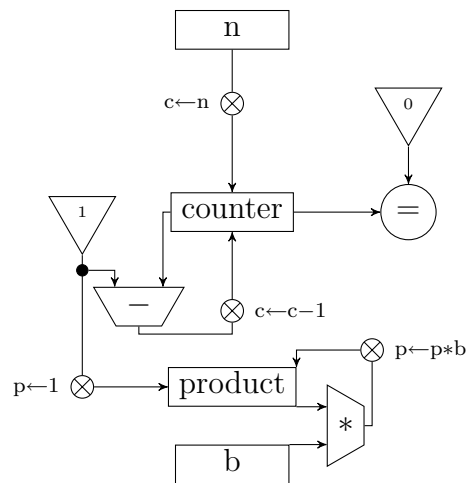
```
( controller
    ( assign  counter  ( reg  n ) )
    ( assign  product  ( const  1 ) )
 test−counter
    ( test  ( op  =)  ( reg  counter )  ( const  0 ) )
    ( branch  ( label  expt−done ) )
    ( assign  counter  ( op  −)  ( reg  counter )  ( const  1 ) )
    ( assign  product  ( op  ∗)  ( reg  product )  ( reg  b ) )
    ( goto  ( label  test−counter ) )
 expt−done )
```