

```

,,
input, output :
gotocacheinput
write, read :
read1 :
read2 :
Buffer
CBufferState == [ cache : ; size: ; ringsize : ; top : ; bot : ring : ]
ControllerInit size := 0 bot := 1 top := 1
CacheInput x : size := 0 size := 1 cache := x
StoreInput (0 ; size) (size := size + 1 top := (top 2) + 1) (0 ; size)
StoreInputController (0 ; size  $\wedge$  size < 2)(size := size + 1 top := (top 2) +
1)(0 >= size  $\vee$  size >= 2)

InputController size ; 2 input?x ( (gotocacheinput CacheInput) (write.top!x
StoreInputController) )
CInput size ; 2 input?x (write.top!2 CacheInput ) (write.top!1 StoreInput)
NoNewCache (size = 0) size := 1 (size  $\neq$  0)
StoreNewCachesize := size - 1 bot := (bot 3) + 1
StoreNewCacheController x : (size > 1)(size := size - 1 cache := x bot' =
(bot 2) + 1)(size <= 1)
OutputController output!cache(read.bot?x StoreNewCacheController(x))(read.bot.3 NoNewCache)
COutput size > 0 output!cache(read.bot.4 StoreNewCache)(read.bot.5 NoNewCache)

ControllerAction ControllerInit ( X ((InputController OutputController)
X))

StoreRing write?i?x
NewCacheRing read?i!2

RingAction X ((StoreRing NewCacheRing) X)
(ControllerAction { size, ringsize, cache, top, bot } — write, read — { ring
} RingAction)

```