

channel*input, output* : \mathbb{N}

channel*gotocacheinput*

channel*write, read* : $\mathbb{N} \times \mathbb{N}$

channel*read1* : \mathbb{N}

channel*read2* : \mathbb{N}

process *Buffer* $\hat{=}$ **begin**

state *CBufferState* $\hat{=}$ [*cache* : \mathbb{N} ; *size* : \mathbb{N} ; *ringsize* : \mathbb{N} ; *top* : \mathbb{N} ; *bot* : \mathbb{N} ; *ring* : $\mathbb{N} \times \mathbb{N}$]

ControllerInit $\hat{=}$ *size* := 0 ; *bot* := 1 ; *top* := 1

CacheInput $\hat{=}$ *x* : \mathbb{N} • *size* := 0 ; *size* := 1 ; *cache* := *x*

StoreInput $\hat{=}$ **if**(0 < *size*) \rightarrow (*size* := *size* + 1 ; *top* := (*top* mod 2) + 1) **||** (0 >= *size*) \rightarrow *Stop*

StoreInputController $\hat{=}$ **if**(0 < *size* \wedge *size* < 2) \rightarrow (*size* := *size* + 1 ; *top* := (*top* mod 2) + 1) **||** (0

InputController $\hat{=}$ \langle *size* < 2 \rangle & *input*?*x* \rightarrow ((*gotocacheinput* \rightarrow *CacheInput*) \square (*write.top*!*x* \rightarrow *S*

CInput $\hat{=}$ \langle *size* < 2 \rangle & *input*?*x* \rightarrow (*write.top*!2 \rightarrow *CacheInput*) \square (*write.top*!1 \rightarrow *StoreInput*)

NoNewCache $\hat{=}$ **if**(*size* = 0) \rightarrow *size* := 1 **||** (*size* \neq 0) \rightarrow *Stop*

StoreNewCache $\hat{=}$ *size* := *size* - 1 ; *bot* := (*bot* mod 3) + 1

StoreNewCacheController $\hat{=}$ *x* : \mathbb{N} • **if**(*size* > 1) \rightarrow (*size* := *size* - 1 ; *cache* := *x* ; *bot*' = (*bot* mod 3) + 1)

OutputController $\hat{=}$ *output*!*cache* \rightarrow (*read.bot*?*x* \rightarrow *StoreNewCacheController*(*x*)) \square (*read.bot*!2 \rightarrow *StoreNewCacheController*(*x*))

COutput $\hat{=}$ \langle *size* > 0 \rangle & *output*!*cache* \rightarrow (*read.bot*.4 \rightarrow *StoreNewCache*) \square (*read.bot*.5 \rightarrow *NoNewCache*)

ControllerAction $\hat{=}$ *ControllerInit* ; (μ *X* • ((*InputController* \square *OutputController*) ; *X*))

StoreRing $\hat{=}$ *write*?*i*?*x* \rightarrow *Skip*

NewCacheRing $\hat{=}$ *read*?*i*!2 \rightarrow *Skip*

RingAction $\hat{=}$ μ *X* • ((*StoreRing* \square *NewCacheRing*) ; *X*)

• (*ControllerAction* **||** { *size, ringsize, cache, top, bot* } | { *write, read* } | { *ring* } **||** *RingAction*)

end