

РЕШЕЊЕ ЈУН АРХИТЕКТУРА РАЧУНАРА

K2

3. Задатак

а)

Р.Б.	Адресни простор	Почетна адреса	Последња адреса
1.	Адресни простор резервисан за RAM	4000 0000	FFFF FFFF
2.	Адресни простор резервисан за ROM	0000 0000	3FFF FFFF
3.	Улазно/излазни адресни простор	F000 0000	FFFF FFFF
4.	Физичка RAM меморија	4000 0000	7FFF FFFF
5.	Физичка ROM меморија	0000 0000	1FFF FFFF
6.	Улазно/излазни адресни простор попуњен	FFFF FFF0	FFFF FFFF
Р.Б.	Улазно/излазни адресни простор	Почетна адреса	Последња адреса
1.	Улазно/излазни адресни простор	000	FFF
2.	Улазно/излазни адресни простор попуњен	FF0	FFF

K3 - 4. Задатак

```
void sendDataSync(int* blok, int size)
```

endDataSync:

```
PUSH          ; чување акумулатора
PUSHBP        ; чување старог BP
LOAD SP       ;
STORE BP      ; BP<=SP
```

high	size
	*block
	PC
	ACC
low	BP
	ENDBLOCK

```
loop1:IN FF00h    ; чекам да се периферија не користи
TST 1h           ; тест маске
JZ loop1         ; ако је периферија заузета, провери поново
```

```
LOAD [BP+4]      ; локална променљива ENDBLOCK = адреса последњег податка +1
ADD [BP+3]
PUSH
```

```
LOAD [BP+3]
STORE GLOB0      ; GLOB0 = адреса податка за слање
```

```
LOAD #1          ; покрећем периферију за слање
OUT FF00h
```

```
nextData:LOAD [BP-1]; ACC <= ENDBLOCK
CMP GLOB0
JZ endSendDataSync ; провера да ли је послат цео блок
```

```
loop2: IN FF01h    ; провера Ready бита
TST 1h
JZ loop2
```

```
LOAD [GLOB0]     ; шаљемо податак на периферију
OUT FF02h
```

```
INC GLOB0        ; померамо адресу на следећи податак
```

```
JMP nextData
```

```
LOAD #0
OUT FF00h        ; гашење периферије
```

```
LOAD BP
STORE SP
POP BP
POP
RTS
```

```
void sendDataAsync(int* blok, int size)
```

```
sendDataAsync:
```

```
PUSH          ; чување акумулатора
PUSHBP        ; чување старог BP
LOAD SP       ;
STORE BP      ; BP<=SP

loop1:IN FF00h ; чекам да се периферија не користи
TST 1h        ; тест маске
JZ loop1      ; ако је периферија заузета, провери поново

LOAD [BP+4]    ; GLOB1 = адреса последњег податка +1
ADD [BP+3]
STORE GLOB1

LOAD [BP+3]
STORE GLOB0    ; GLOB0 = адреса податка за слање

LOAD #1
STORE ASYNC_SEND ; почиње пренос података

LOAD #81h      ; палим периферију за слање и прекидом
OUT FF00h

LOAD BP
STORE SP
POP BP
POP
RTS
```

прекидна рутина PER0:

```
PUSH

LOAD [GLOB0]   ; шаљемо податак на периферију
OUT FF02h

INC GLOB0      ; померамо адресу на следећи податак

LOAD GLOB1     ; ACC <= последња адреса податка + 1
CMP GLOB0
JNZ end        ; провера да ли је послат цео блок

LOAD #0
STORE ASYNC_SEND
OUT FF00h

end:POP
RTI
```

high	size
	*block
	PC
	ACC
low	BP