

# ЛПРС2 ММ Лаб

Меморијско Мапирање и Улаз-Излаз

верзија 1.0

Милош Суботић

8. март 2021.

## 1 Увод

Циљ вежбе је провежбати прозивку и прекиде над периферној јединици. и као и научити радити са стандардном периферном јединицом тајмера.

## 2 Емулатор

Пројекат се ради са WAF. Први пут на неком рачунару, ради инсталирања потребних библиотак, покренути:

```
1 ./waf prerequisites
```

Водити рачуна да када копи-пејстујете код из PDF-а буду додати размаци на све стране, па их треба побрисати.

Када први пут покрећете отпакован или потпуно очишћен пројекат, потребно га је исконфигурисати са:

```
1 ./waf configure
```

Ова команда ће пронаћи компајлер и библиотеке неопходне за компајлирање пројекта. У случају Windows-а, потребно је да WAF пронађе GCC и G++ компајлер од MinGW-а из MSYS2 окружења. У случају да је Visual Studio инсталиран, WAF може да повуче MSVC као компајлер, што неће радити после у компајлирању. У том случају покренути конфигурацију са додатним опцијама:

```
1 ./waf configure --check-c-compiler=gcc --check-cxx-compiler=g++
```

Ове опције ће натерати WAF да користи MinGW компајлере из MSYS2 окружења.

Након стандардне конфигурације компајлирање (билдовање) и покретање одговарајуће апликације се може постићи са:

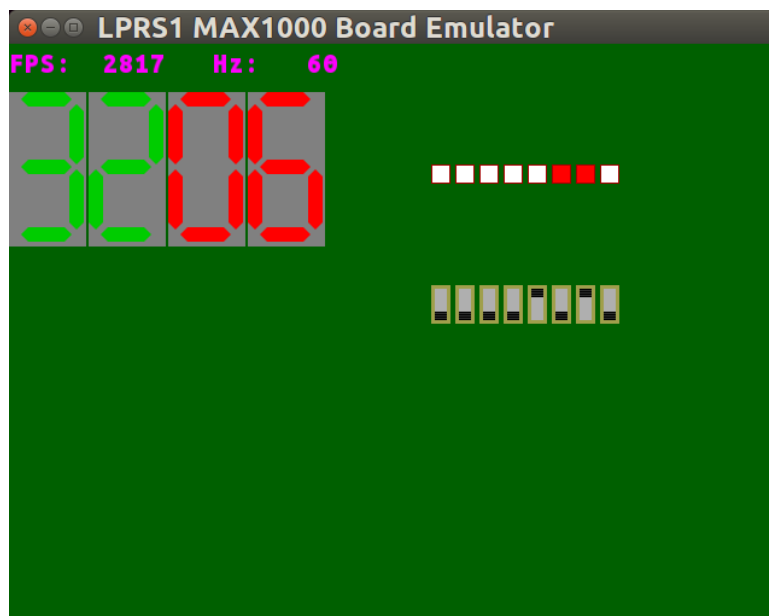
```
1 ./waf build run --app=app_01_ptr
```

Ако ова команда не ради, компајлирање урадити са:

```
1 ./waf build
```

а онда покретање са:

```
1 ./build/app_01_ptr
```



Слика 1: Изглед графичке корисничке спреге емулятора

Након покретања биће приказана графичка корисничка спрега као на Слици 1. Емулирани елементи су:

- 7-сегментни дисплеји,
- ледаче,
- прекидачи.

Њихов распоред је стандардан: LSB је скроз десни. **Прекидачи се контролишу са притиском бројке на тастатури.** На пример, у горњој апликацији, MSB тј. скроз леви тј. 7. прекидач је постављен надоле тј. логичка 0. Притиском на тастер 7 на тастатури горепоменути тастер ће бити пребачен на логичку 1. Понован притисак на тастер га пребацује на логичку 0, итд.

### 3 Задаци

Користити `busy_wait` као извор тајминга. Табела 1 је легенда шта значе одређене скраћенице у спецификацији.

Табела 1: Легенда спецификације

32	преко 32-битног показивача (индексирање, макрои, маскирање)
bf	преко битског поља
p	користити паковане регистре
u	користити распаковане регистре

#### 3.1 *app\_04\_stopwatch*

Израдити штоперицу. Користити низ од 4 бајта за софтверско бројање цифара штоперице. Табела 2 дефинише која периферија има коју функцију: За излазне шта треба

приказати на њима, а за улазне на шта утичу. Табела 3 дефинише који студент треба да користи коју јединицу на који начин.

Табела 2: Функције периферија *app\_04\_stopwatch*

Периферија	Регистар	Функција
7-сегменти дисплеји	SEGM0	Јединице стотинке
	SEGM1	Десетице стотинке
	SEGM2	Јединице секунде
	SEGM3	Десетице секунде
Ледаче	LED[3:0] LED[7:4]	Десетице стотинке Јединице секунде
Прекидачи	SW0	Ресет
	SW1	Пауза

Табела 3: Спецификација за *app\_04\_stopwatch*

Индекс бити [2:0]	000	001	010	011	100	101	110	111
SW	bf p	32 u	bf u	32 p	bf p	32 u	bf u	32 p
LED	32 u	bf p	32 p	bf u	32 u	bf p	32 p	bf u
7SEGM	bf p	32 u	bf p	32 u	bf u	32 p	bf u	32 p

### 3.2 *app\_05\_chess\_clock*

Израдити сат за шах. Користити 2 низа од по 2 бајта за софтверско бројање цифара. Табела 4 дефинише која периферија има коју функцију: За излазне шта треба приказати на њима, а за улазне на шта утичу. Табела 5 дефинише који термин треба да користи коју јединицу на који начин.

Табела 4: Функције периферија *app\_05\_chess\_clock*

Периферија	Регистар	Функција
7-сегменти дисплеји	SEGM0	Јединице секунде белог играча
	SEGM1	Десетице секунде белог играча
	SEGM2	Јединице секунде црног играча
	SEGM3	Десетице секунде црног играча
Ледаче	LED0 LED7	Активан бели играч Активан црни играч
Прекидачи	SW0	Ресет
	SW1	Пауза
	SW7	Активан: 0 - бели, 1 - црни

Табела 5: Спецификација за *app\_05\_chess\_clock*

Индекс бити [2:0]	000	001	010	011	100	101	110	111
SW	32 u	bf p	32 p	bf u	32 u	bf p	32 p	bf u
LED	bf p	32 u	bf u	32 p	bf p	32 u	bf u	32 p
7SEGM	32 u	bf p	32 u	bf p	32 p	bf u	32 p	bf u

## 4 Регистарске мапе

Слика 4 приказује регистарску мапу PI0 периферије. Читајући "0 (0x00)", може се закључити да је то 0-ти регистар, са **релативном адресом** 0x00 (релативном унутар тог периферије). Само LSB бит је мапиран на 0. бит ледаче, док су остали виши бити ([31:1]) **резервисани** (назначено са -). Регистар LED0 дозвољава читање и писање **Read-Write**. Приликом читања овог регистра, прочитаће се оно што је у њега претходно записано тј. његова почетвна вредност ако ништа до тада није било писано у њега. Резервисани бити се могу само читати, док се писање у њих игнорише. Слично се и 8. регистар може само читати. То су такозвани **Read-Only** регистри. У првих 16 регистра су ледаче и прекидачи у расутим **unpack** формату, док су у 16. и 17. регистру ледаче и прекидачи у пакованом **pack** формату. Почетна вредност резервисаних бита је 0 и то ће бити резултат читање тих бита. Почетна вредност за прекидаче није од великог значаја, јер након пуштања ресета, ти бити ће имати вредност прекидача.

Bits	31		1	0
0 (0x00)		-		LED0
Read/Write		R		RW
Initial Value		0		1
Bits	31		1	0
1 (0x04)		-		LED1
Read/Write		R		RW
Initial Value		0		0
Bits	31		1	0
2 (0x08)		-		LED2
Read/Write		R		RW
Initial Value		0		1
Bits	31		1	0
3 (0x0c)		-		LED3
Read/Write		R		RW
Initial Value		0		0
Bits	31		1	0
4 (0x10)		-		LED4
Read/Write		R		RW
Initial Value		0		0
Bits	31		1	0
5 (0x14)		-		LED5
Read/Write		R		RW
Initial Value		0		0
Bits	31		1	0
6 (0x18)		-		LED6
Read/Write		R		RW
Initial Value		0		0
Bits	31		1	0
7 (0x1c)		-		LED7
Read/Write		R		RW
Initial Value		0		0

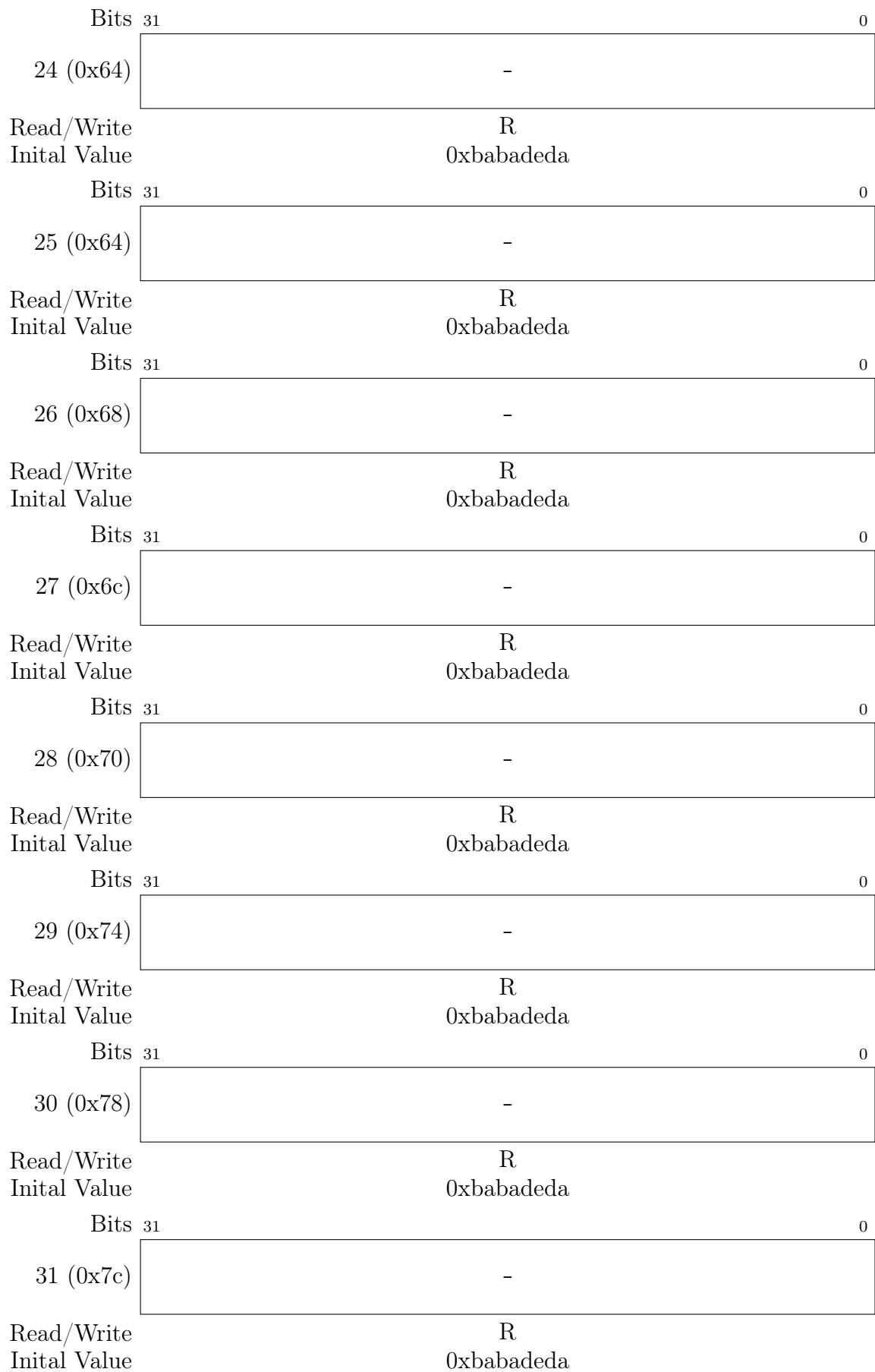
Слика 2: PIO меморијска мапа 1/4

Bits	31	1	0
8 (0x20)	-	SW0	
Read/Write	R	R	
Initial Value	0	0	
Bits	31	1	0
9 (0x24)	-	SW1	
Read/Write	R	R	
Initial Value	0	1	
Bits	31	1	0
10 (0x28)	-	SW2	
Read/Write	R	R	
Initial Value	0	0	
Bits	31	1	0
11 (0x2c)	-	SW3	
Read/Write	R	R	
Initial Value	0	1	
Bits	31	1	0
12 (0x30)	-	SW4	
Read/Write	R	R	
Initial Value	0	0	
Bits	31	1	0
13 (0x34)	-	SW5	
Read/Write	R	R	
Initial Value	0	0	
Bits	31	1	0
14 (0x38)	-	SW6	
Read/Write	R	R	
Initial Value	0	0	
Bits	31	1	0
15 (0x3c)	-	SW7	
Read/Write	R	R	
Initial Value	0	0	

Слика 3: PIO меморијска мапа 2/4

16 (0x40)	Bits 31	8	7	0
		-	LED[7:0]	
Read/Write		R	RW	
Initial Value		0	0x05	
17 (0x44)	Bits 31	8	7	0
		-	SW[7:0]	
Read/Write		R	R	
Initial Value		0	0x0a	
18 (0x48)	Bits 31			0
		-		
Read/Write		R		
Initial Value		0xbabadedda		
19 (0x4c)	Bits 31			0
		-		
Read/Write		R		
Initial Value		0xbabadedda		
20 (0x50)	Bits 31			0
		-		
Read/Write		R		
Initial Value		0xbabadedda		
21 (0x54)	Bits 31			0
		-		
Read/Write		R		
Initial Value		0xbabadedda		
22 (0x58)	Bits 31			0
		-		
Read/Write		R		
Initial Value		0xbabadedda		
23 (0x5c)	Bits 31			0
		-		
Read/Write		R		
Initial Value		0xbabadedda		

Слика 4: PIO меморијска мапа 3/4



Слика 5: PIO меморијска мапа 4/4



Bits	31								4	3	0									
0(0x00)	-									SEGM0										
Read/Write	R									RW										
Initial Value	0									0										
Bits	31								4	3	0									
1(0x04)	-									SEGM1										
Read/Write	R									RW										
Initial Value	0									1										
Bits	31								4	3	0									
2(0x08)	-									SEGM2										
Read/Write	R									RW										
Initial Value	0									2										
Bits	31								4	3	0									
3(0x0c)	-									SEGM3										
Read/Write	R									RW										
Initial Value	0									3										
Bits	31					16	15		12	11		8	7		4	3	0			
4(0x10)	-				SEGM3				SEGM2				SEGM1				SEGM0			
Read/Write	R				RW				RW				RW				RW			
Initial Value	0				3				2				1				0			

Слика 6: TIME\_MUXED\_7SEGM меморијска мапа