

ЦИФРОВА СХЕМОТЕХНИКА

ЛЕКЦИЯ #15

Процесор PicoBlaze. Архитектура, инструкции. Използване при изграждане на проекти в среда Xilinx ISE WebPack

Основни характеристики – предимства, ограничения

- ❑ структура от типа “**soft core**” /cell-level HDL description/ за използване (влагане) в Xilinx чипове;
- ❑ проектиран с наличните ресурси от чипа и с възможност за използване едновременно с останалите структури от FPGA;
- ❑ заема само около 200 логически клетки (под 10% от ресурсите на най-малкия 3S100 чип от фамилията Spartan3E);
- ❑ ограничени възможности за обработка с висока скорост (low speed performance);
- ❑ с приложение основно за I/O операции и управление на некритични по отношение на бързодействие приложения.

Всяка инструкция се изпълнява в рамките на 2 машинни цикъла – т.е. при CLK=100MHz → 50 мил. инструкции/сек (50 MIPS);

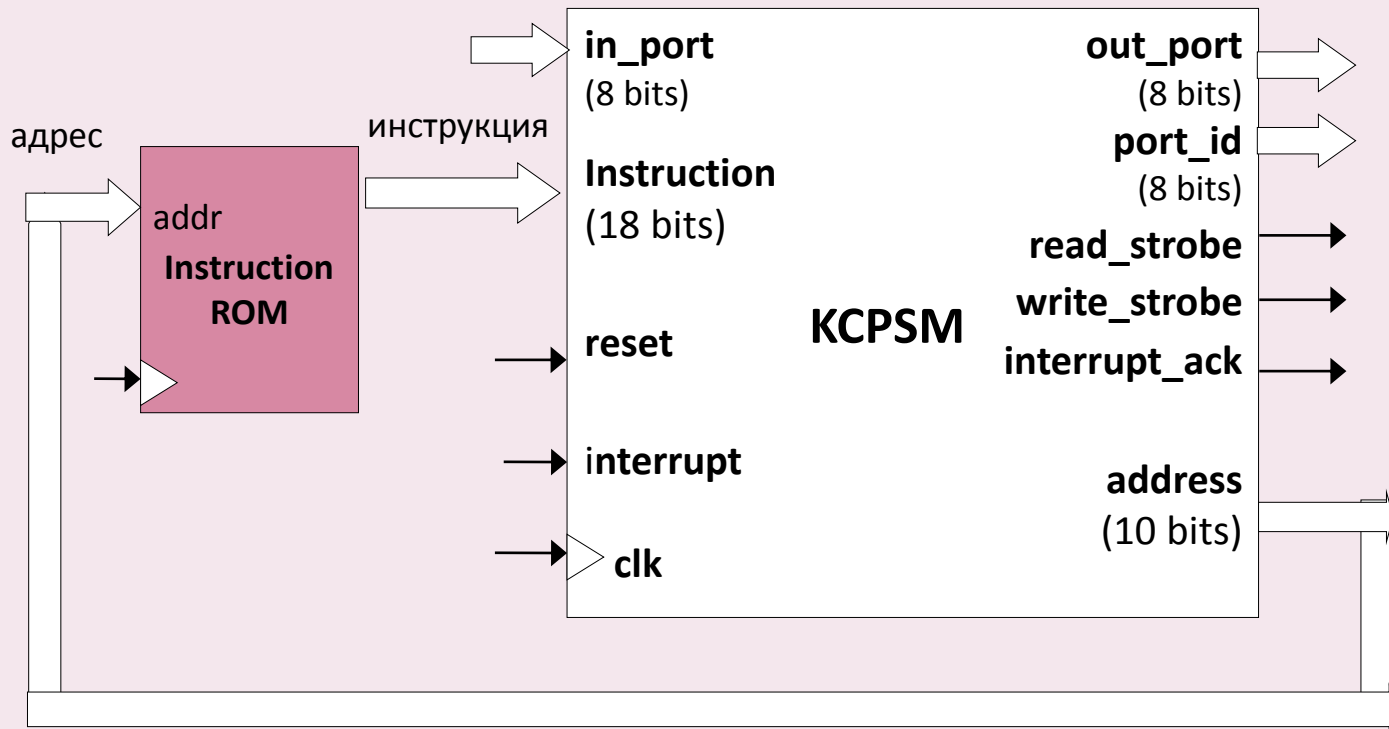
Процесор PicoBlaze. Архитектура, инструкции. Използване при изграждане на проекти в среда Xilinx ISE WebPack

Общи характеристики на процесора:

- 8-битова дължина на данните;
 - АЛУ с флагове само за пренос и нула (carry, zero);
 - 16 броя 8-битови регистри с общо предназначение;
 - 64 байта памет за данни (data RAM);
 - адрес на инструкциите – 10 бита (т.е. програма с до 1024 инструкции);
 - дължина на инструкциите – 18 бита;
 - стек с дълбочина 31 думи;
 - 256 входни и 256 изходни порта;
 - блок за управление на прекъсванията.
-
- ❖ 2 цикъла за изпълнение на инструкция;
 - ❖ 5 цикъла за обработка на прекъсване.

Процесор PicoBlaze. Архитектура, инструкции. Използване при изграждане на проекти в среда Xilinx ISE WebPack

Модулна HDL организация (два top-level модули)



K(c)onstant **C**oded **P**rogrammable **S**tate **M**achine (KCPSM)
/оригиналното наименование на процесора PicoBlaze/

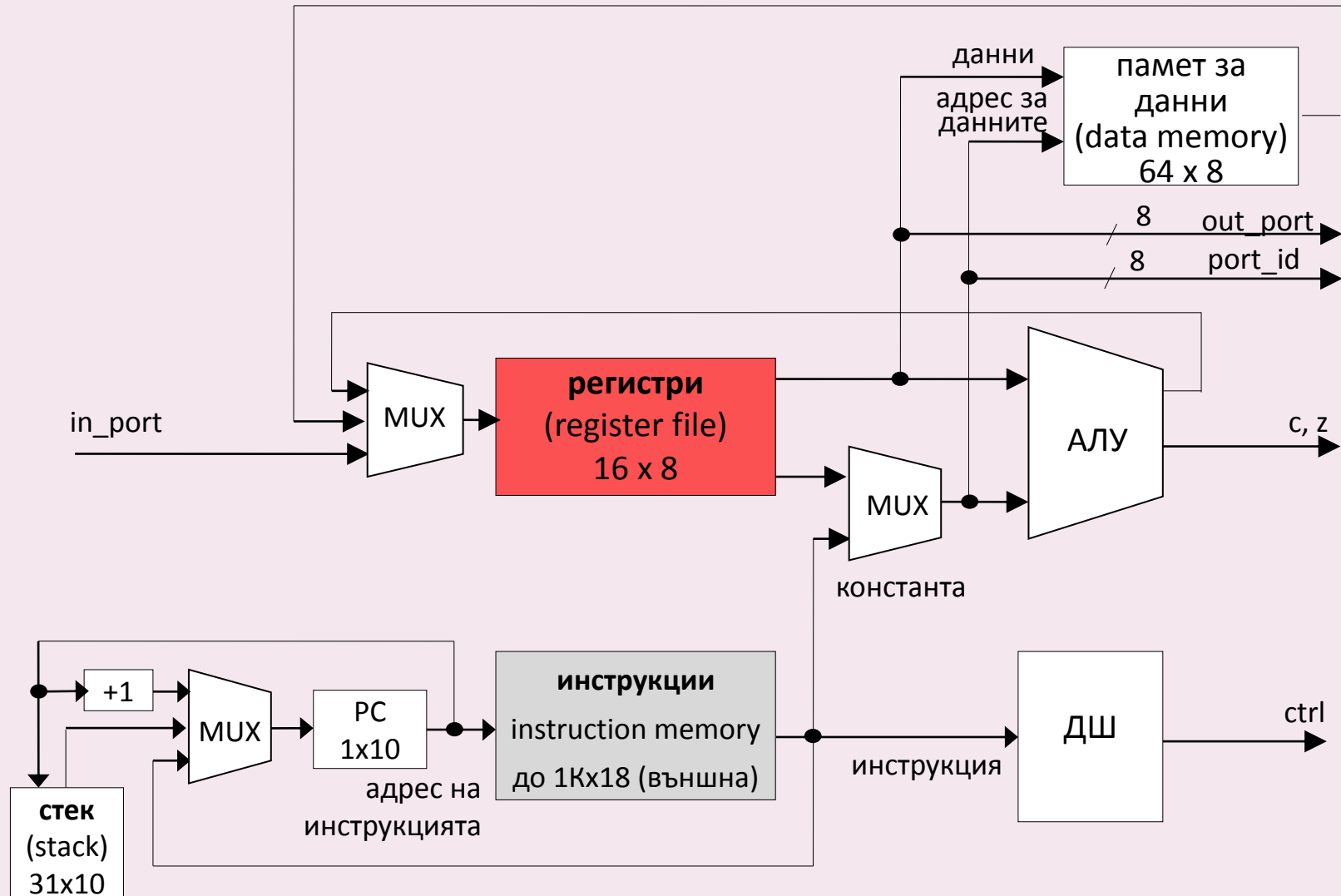
Процесор PicoBlaze. Архитектура, инструкции. Използване при изграждане на проекти в среда Xilinx ISE WebPack

❖ Входно/изходни сигнали за процесора:

- **clk** – системен тактов сигнал;
- **reset** – нулиране/нач.установяване;
- **address** (10 бита) – адрес към паметта за инструкции (до 1К байта);
- **instruction** (18 бита) – инструкция;
- **port_id** (8 бита) – адрес на I/O порт (при четене/запис от периф.устройства);
- **in_port** (8 бита) – входни данни от периферни устройства;
- **read_strobe** – строб за входните операции;
- **out_port** – изходни данни към периферни устройства;
- **write_strobe** – строб за изходните операции;
- **interrupt** – заявка за прекъсване от I/O устройство;
- **interrupt_ack** – потвърждение за обработка на прекъсване към I/O устройство.

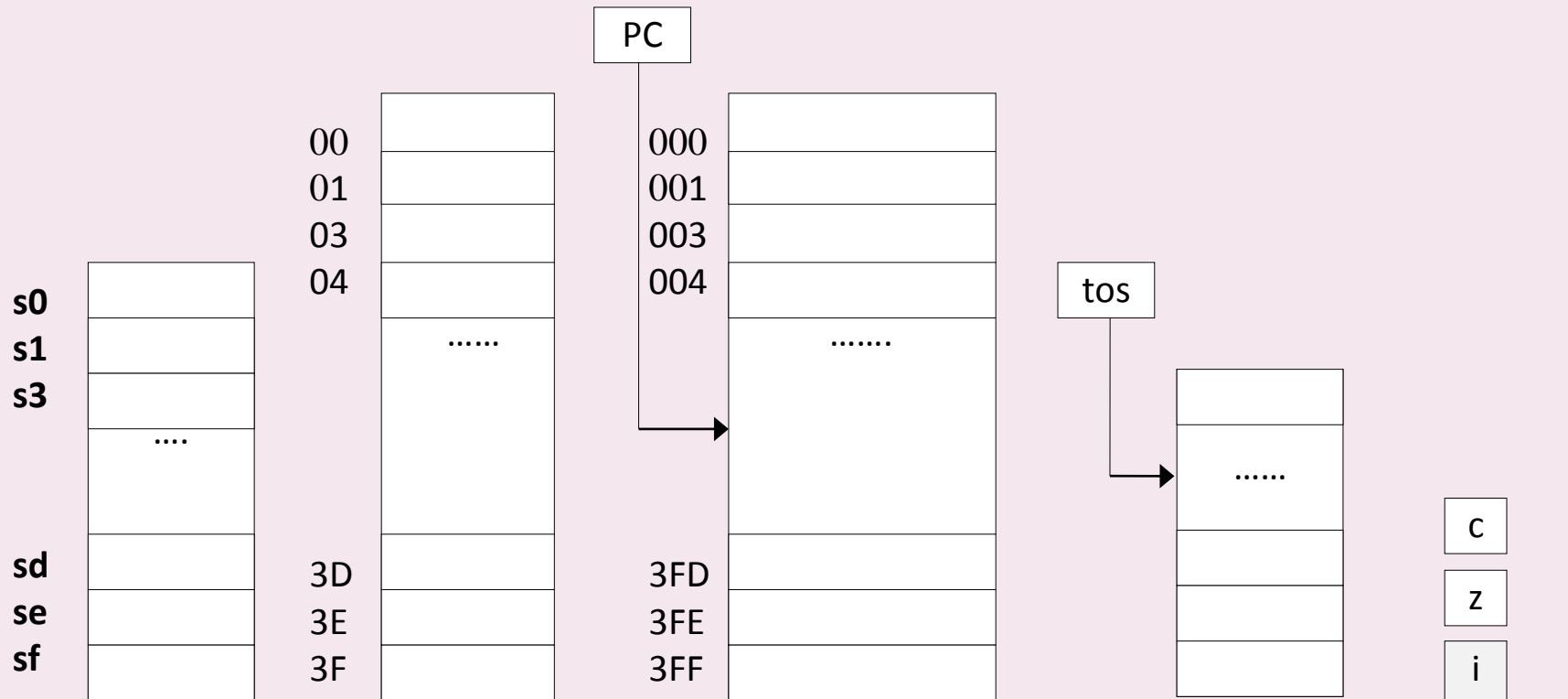
Процесор RicoBlaze. Архитектура, инструкции. Използване при изграждане на проекти в среда Xilinx ISE WebPack

○ Вътрешна архитектура:



Процесор PicoBlaze. Архитектура, инструкции. Използване при изграждане на проекти в среда Xilinx ISE WebPack

Програмен модел



регистри с общо предн.

register file

/16 броя/

RAM за данни

data RAM

/64 bt/

памет за инструкции

instruction memory (ROM)

/1024 bt → 1K/

стек

call/return stack

/0-31/

флагове

flags

2+1

Процесор RicoBlaze. Архитектура, инструкции. Използване при изграждане на проекти в среда Xilinx ISE WebPack

Обобщен формат на инструкциите на процесора

- Тип “Регистър-регистър”: **КОП** sX,sY /sX ← sX КОП sY/
- Тип “Регистър-константа”: **КОП** sX,КК /sX ← sX КОП КК/
- Тип “Единичен регистър”: **КОП** sX /sX ← КОП sX/
- Тип “С единичен адрес”: **КОП** ААА
- Тип “С нулев операнд”: **КОП**

Процесор RicoBlaze. Архитектура, инструкции. Използване при изграждане на проекти в среда Xilinx ISE WebPack

Набор инструкции на процесора (видове, синтаксис)

- **аритметични:** **add** sX,sY /sX \leftarrow sX + sY/; **add** sX,KK /sX \leftarrow sX + KK/
addcy sX,sY /sX \leftarrow sX + sY + c/; **addcy** sX,KK /sX \leftarrow sX + KK + c/
>> аналогично – команди **sub**, **subcy**
- **логически:** **and** sX,sY /sX \leftarrow sX and sY/; **and** sX,KK /sX \leftarrow sX and KK/
>> аналогично – команди **or**, **xor**
- **за сравнение:** **compare** sX,sY; **compare** sX,KK
и тестови (при sX=sY z \leftarrow '1'; при sY>sX c \leftarrow '1')
test sX,sY; **test** sX,KK;
(t \leftarrow sX and sY ; при t=0 z \leftarrow '1')

Процесор RicoBlaze. Архитектура, инструкции. Използване при изграждане на проекти в среда Xilinx ISE WebPack

Набор инструкции на процесора (видове, синтаксис)

■ за преместване / ротиране

(4 за преместване наляво, 4 за надясно, 2 за ротиране):

sl0 sX /премества sX 1 бит наляво, т.е. $sX(0) \leftarrow '0'$, $c \leftarrow sX(7)$ /

sl1 sX /премества sX 1 бит наляво, т.е. $sX(0) \leftarrow '1'$, $c \leftarrow sX(7)$ /

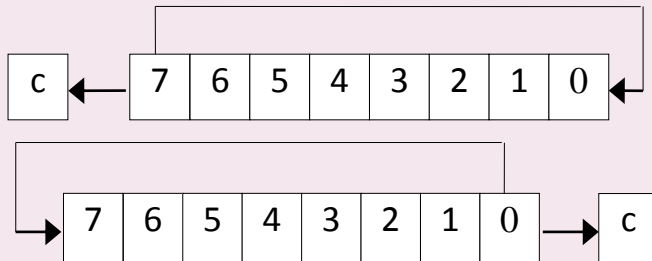
slx sX /премества sX 1 бит наляво, запазва sX(0), $c \leftarrow sX(7)$ /

sla sX /премества sX 1 бит наляво, $sX(0) \leftarrow 'c'$, $c \leftarrow sX(7)$ /

>> аналогично – команди: **sr0** sX, **sr1** sX, **srx** sX, **sra** sX (за надясно)

rl sX /ротира регистъра sX с 1 бит в ляво/

rr sX /ротира регистъра sX с 1 бит в дясно/



Процесор PicoBlaze. Архитектура, инструкции. Използване при изграждане на проекти в среда Xilinx ISE WebPack

Набор инструкции на процесора (видове, синтаксис)

■ **за преместване на данни.** Видове:

- *между регистри* - инструкции **load**;
- *между регистър и памет за данни (data RAM)* - инструкции **fetch, store**;
- *между регистър и I/O порт* - инструкции **input, output**.

Примери: **load** sX,sY /sX←sY/

fetch sX,sY /sX←RAM[sY]/

store sX,sY /RAM[sY]←sX/

input sX,sY /port_id←sY; sX←in_port/ алтерн. **in** sX,sY

output sX,sY /port_id←sY; out_port←sX/ алтерн. **out** sX,sY

Процесор PicoBlaze. Архитектура, инструкции. Използване при изграждане на проекти в среда Xilinx ISE WebPack

Набор инструкции на процесора (видове, синтаксис)

■ за контрол изпълнението на програмата.

Управлява се принудително стойността на PC (в противен случай $PC=PC+1$):

jump AAA /PC←AAA/ -- безусловен преход на адрес AAA
jump c,AAA /PC←AAA/ -- условен преход на адрес AAA само при c='1'
jump nc,AAA /PC←AAA/ -- условен преход на адрес AAA само при c='0'
jump z,AAA /PC←AAA/ -- условен преход на адрес AAA само при z='1'
jump nz,AAA /PC←AAA/ -- условен преход на адрес AAA само при z='0'

call – изпраща PC на конкретен адрес за обработка на процедура:

call AAA -- безусловно извикване на процедура на адрес AAA
call c,AAA -- извикване на процедура на адрес AAA само при c='1'
call nc,AAA -- извикване на процедура на адрес AAA само при c='0'
call z,AAA -- извикване на процедура на адрес AAA само при z='1'
call nz,AAA -- извикване на процедура на адрес AAA само при z='0'

>> Аналогични процедури за **return**. Връща PC на адрес +1 от предходния, съхранен в стека.

Процесор RicoBlaze. Архитектура, инструкции. Използване при изграждане на проекти в среда Xilinx ISE WebPack

Набор инструкции на процесора (видове, синтаксис)

■ за обработка на прекъсвания:

За разлика от jump и call/return – идват от външна заявка (request).

При това програмният брояч се зарежда с $PC \leftarrow 3FF$ (вектор за обработка на прекъсвания).

returni disable (reti disable*) -- връщане от сервизната процедура и флагът $i \leftarrow '0'$

returni enable (reti enable*) -- връщане от сервизната процедура и флагът $i \leftarrow '1'$

disable interrupt (dint*) -- забранява заявка за прекъсване $i \leftarrow '0'$

enable interrupt (eint*) -- разрешава заявка за прекъсване $i \leftarrow '1'$

* Инструкции за асемблера на Mediatronix.

Процесор PicoBlaze. Архитектура, инструкции. Използване при изграждане на проекти в среда Xilinx ISE WebPack

□ Директиви на процесора:

Не са част от набора инструкции.

Цел – да “накарат” процесора да изпълни специфични операции (дефиниране на константа, резервиране на място за данни).

Приложение – при изграждане кода на програмата.

▪ Директиви за асемблера KCPSM3 (на Xilinx):

- **address** XXX -- разполага кода на програмата на адрес XXX
- **namereg** s3, <име> -- дава име на определен регистър s3
- **constant** KR, <име> -- дава име на определена константа KR

Процесор PicoBlaze. Архитектура, инструкции. Използване при изграждане на проекти в среда Xilinx ISE WebPack

■ Директиви на асемблера PBlazeIDE (на Mediatronix):

org -- разполага кода на програмата на конкретен адрес в ROM

Пример: **org** \$3FF

equ -- дава име на регистър или на изчислена стойност

Примери: pro_d **equ** s3
res_temp **equ** 256/15

dsin, dsout, dsio -- дават име на I/O портове със специфичен адрес

Пример: keyboard **dsin** \$0E

vhdl -- преобразува (генерира) асемблерските инструкции (instruction ROM) във VHDL формат.

Процес на изграждане на проекти с използване на PicoBlaze (Design Flow)

