**ERM16**

**ERM16 - Easy RISC Microprocessor 16 bits.**

**простой с упрощенным набором инструкций микропроцессор 16 битный.**

**разрядность: 16 бит**

**word : 16 bits**

**тип набор команд : RISC**

**тип конструкции процессора: многотактный**

**набор инструкций кол-во: 41**

**порты:**

**входные :**

**clk : тактовый сигнал**

**VDD : logic 1**

**GND: logic 0**

**DI: шина данных 16 битная**

**выходные:**

**IOE : сигнал делающий запрос к устройствам ввода ввывода к их памяти**

**INTREQ : сигнал оповещающий контроллер шины о том что идет запрос на обработку прерывание перефрийным устройством**

**addr\_bus : адрессная 16 битная шина**

**wrmem : сигнал разрешения записи в память**

**DO : 16 битная шина данных**

**набор иструкций:**

**commands:**

**ldw (load word) ldw rd,[r] загрузка слова из памяти в регистр,адресация только регистровая через любой регистр общего назначения а также через стек поинтер (sp)**

**stw (store word) stw rs,[r] сохранить 16 битное значение из регистра в память ,адресация только регистровая через любой регистр общего назначения а также через стек поинтер (sp)**

**mov (move) mov rd,rs/imm6 скопировать значение из одного регистра в другой или загрузить не посредственное 6 битное значение в регистр**

**alr (addreass load register) alr rd,addr16 загрузка адреса в регитср общего назначения или sp**

**j (jump) j addr16 безусловный переход**

**jz (jump if zero) jz addr16 условный переход если ZF = 1**

**jc (jump if carry) jc addr16 условный переход если CF = 1**

**jo (jump if oveflow) jo addr16 условный переход если OF = 1**

**jp (jump if parity) jp addr16 условный переход если PF = 1**

**jnz (jump if no zero) jnz addr16 условный переход если ZF = 0**

**jnc (jump no carry) jnc addr16 условный переход если CF = 0**

**jno (jump no overflow) jno addr16 условный переход если OF = 0**

**jnp (jump no parity) jnp addr16 условный переход если PF = 0**

**jg (jump if greater) jg addr16 условный переход если GF = 1**

**jl (jump if lower) jl addr16 условный переход если LF = 1**

**jge (jump if greater or equal) jge addr16 условный переход если ZF = 1 or GF = 1**

**jle (jump if lower or equal) jle addr16 условный переход если LF = 1 or ZF = 1**

**ret (return) ret возврат из функции, процессор извлекает из регистра RAF и изменяет PC**

**ch (call handler) ch addr16 вызов функции/процедуры/подпрограммы перемещает pc на addr и сохраняет адрес возврата(следующей инструкции) в RAF**

**int (interrupt) int vector8 запрос аппаратного прерывания к устройству по следством вектора**

**rst (reset cpu) rst сброс процессора в состояние инициализации**

**hlt (halt cpu) hlt остановить процессор**

**add (+) add rd,rs. rd = rd + rs + ci**

**sub (-) sub rd,rs. rd =(rd – rs) - ci**

**mul (\*) mul rd,rs. Rd = rd\*rs**

**div (/) div rd,rs. Rd = rd/rs**

**lsl (<<) lsl rd,rs/imm4 . rd = rd << rs/imm4**

**asr (>>>) asr rd,rs/imm4 . rd = rd >>> rs,imm4**

**lsr (>>) lsr rd,rs/imm4 . rd = rd >> rs,imm4**

**or - or rd,rs/imm8. Rd = rd | rs/imm6**

**eor (exlusive or) eor rd,rs/imm6. Rd = rd ^ rs**

**orn (or not) orn rd,rs/imm6. Rd = rd ~| rs**

**and – and rd,rs/imm6. Rd = rd & rs**

**andn (and not) and rd,rs/imm6. rd = rd ~& rs**

**eon (exlusive or not) eon rd,rs/imm6 . rd = rd ~^ rs**

**mvn (move not) mvn rd,rs/imm6 . rd = ~rs/imm8**

**rev (reverse bits) rev rd,rs. Поменять местами юайты переставить наоборот**

**neg (negative) neg rd,rs. Rd = ~rs+1**

**in (read input port I/O devices) in rd,r/imm6 считать с буфера устройства ввода вывода данные**

**out (write output port I/O devices) out r/imm6,rs записать данные в буфер устройства ввода вывода**

**cmp (compare) cmp rd,rs/imm6 сравнить rd с rs или с imm6**

**префикс команды или расширение:**

**r – register указывает что процессор работает только с регистрами процессора**

**i – immediate указывает что процессор работает и с регистрами и с непосредственными значениями**

**префиксы указываются обязательно во всех инструкциях за исключением ldw stw alr rst hlt int ch ret.**

**пример:**

**movr r2,r5**

**movi r2,0x4f**

**запись инструкции:**

**команда префикс/расширение операнд1,операнд2**

**[15:9] [8:6] [5:3]/[5:0]**

**размер инструкции 16 бит**

**instr r/i op1,op2**

**Регистры процессора:**

**r0-r6 (register) 7 регистров общего назначения**

**pc ( program counter) счётчик инструкций хранит текущий адрес инструкции**

**raf (return addreass function register) регистр адреса возврата функции содержит адрес возврата**