16bit寄存器解码

功能描述：从外设总线上把合适的数据读入模块内部的寄存器里面

相关接口说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 位宽 | 方向 | 说明 |
| per\_dout | 16 | Output | 外设总线数据输出 |
| mclk | 1 | Input | 时钟信号 |
| per\_addr | 14 | Input | 外设地址总线。430是16位单片机，所以解码宽度是16位，最后一位被省略。又因为这是外设地址总线，外设地址最大宽度是32kb，最高位只能为0，所以最高位省略。 |
| per\_din | 16 | Input | 外设数据输入 |
| per\_en | 1 | Input | 外设使能 |
| per\_we | 2 | Input | 外设写使能（为兼容8位外设而设计）  00：读使能  01：写低位使能  10：写高位使能  11：写16位使能 |
| puc\_rst | 1 | Input | 外设复位信号 |

1. 常量定义

1.Parameter [14:0] BASE\_ADDR = 15’h0000

外设寄存器所占用的绝对地址里面重合的部分

2.Parameter DEC\_WD = 8

外设寄存器最大地址的绝对地址所占用的最大宽度。Dma寄存器占用地址从0x0022到0x00f6。所以基地址是0x0000，偏移地址宽度是8位。

寄存器的绝对地址：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 15-DEC\_WD | | | DEC\_WD | | |
| 15 | 14 |  | DEC\_WD | DEC\_WD-1 |  | 0 |
| 0 | BASE\_ADDR（0x0000） | | | Reg\_offset | | |

3．寄存器偏移地址（reg\_offset的具体内容）

|  |  |
| --- | --- |
| DMACTL0 | 22 |
| DMACTL1 | 24 |
| DMA0CTL | E0 |
| DMA0SA | E2 |
| DMA0DA | E4 |
| DMA0SZ | E6 |
| DMA1CTL | E8 |
| DMA1SA | EA |
| DMA1DA | EC |
| DMA1SZ | EE |
| DMA2CTL | F0 |
| DMA2SA | F2 |
| DMA2DA | F4 |
| DMA2SZ | F6 |

独热码编码

DEC\_SZ = 1<<DEC\_WD (DEC\_SZ==128)

BASE\_REG {{DEC\_SZ-1{1’b0}}，1’b1} (前端补127个0，后面补一个1方便后面进行移位操作)

八位地址数据映射到128位的独热码上

独热码解码：

DMACTL0\_D <= （BASE\_REG << DMACTL0）

DMACTL1\_D <= （BASE\_REG << DMACTL1）

DMA0CTL\_D <= （BASE\_REG << DMA0CTL）

DMA0SA\_D <= （BASE\_REG << DMA0SA）

DMA0DA\_D <= （BASE\_REG << DMA0DA）

DMA0SZ\_D <= （BASE\_REG << DMA0SZ）

DMA1CTL\_D <= （BASE\_REG << DMA1CTL）

DMA1SA\_D <= （BASE\_REG << DMA1SA）

DMA1DA\_D <= （BASE\_REG << DMA1DA）

DMA1SZ\_D <= （BASE\_REG << DMA1SZ）

DMA2CTL\_D <= （BASE\_REG << DMA2CTL）

DMA2SA\_D <= （BASE\_REG << DMA2SA）

DMA2DA\_D <= （BASE\_REG << DMA2DA）

DMA2SZ\_D <= （BASE\_REG << DMA2SZ）

把寄存器的值映射到128位的地址空间上

1. 寄存器解码
2. 寄存器片选

regsel = （per\_em）&（per\_addr[13：DEC\_WD-1] == BASE\_ADDR[14:DEC\_WD]）

外设地址：(14位总线)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 13-(15-DEC\_WD)=DEC\_WD-2 | | | DEC\_WD-1 | | | x |
| 0 | 13 |  | DEC\_WD-1 | DEC\_WD-2 |  | 0 |  |

寄存器的绝对地址

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 15-DEC\_WD | | | DEC\_WD | | |
| 15 | 14 |  | DEC\_WD | DEC\_WD-1 |  | 0 |
| 0 | BASE\_ADDR(0x0000) | | | reg\_offset | | |

1. 寄存器的本地寻址

外设地址后（DEC\_WD-1）位 + 后端补0

wire [DEC\_WD-1:0] reg\_addr = {per\_addr[DEC\_WD-2:0], 1'b0};

1. 寄存器操作
2. reg\_dec [DEC\_WD-1:0]

分别判断reg\_addr与寄存器的绝对地址与之前设定的偏移量是否相等，如果相等，则把对应的register\_D赋值给reg\_dec

wire [DEC\_SZ-1:0] reg\_dec = (DMACTL0\_D & {DEC\_SZ{(reg\_addr == DMACTL0 )}}) |

(DMACTL1\_D & {DEC\_SZ{(reg\_addr == DMACTL1 )}}) |

(DMA0CTL\_D & {DEC\_SZ{(reg\_addr == DMA0CTL )}}) |

(DMA0SA\_D & {DEC\_SZ{(reg\_addr == DMA0SA )}}) |

(DMA0DA\_D & {DEC\_SZ{(reg\_addr == DMA0DA )}}) |

(DMA0SZ\_D & {DEC\_SZ{(reg\_addr == DMA0SZ )}}) |

(DMA1CTL\_D & {DEC\_SZ{(reg\_addr == DMA1CTL )}}) |

(DMA1SA\_D & {DEC\_SZ{(reg\_addr == DMA1SA )}}) |

(DMA1DA\_D & {DEC\_SZ{(reg\_addr == DMA1DA )}}) |

(DMA1SZ\_D & {DEC\_SZ{(reg\_addr == DMA1SZ )}}) |

(DMA2CTL\_D & {DEC\_SZ{(reg\_addr == DMA2CTL )}}) |

(DMA2SA\_D & {DEC\_SZ{(reg\_addr == DMA2SA )}}) |

(DMA2DA\_D & {DEC\_SZ{(reg\_addr == DMA2DA )}}) |

(DMA2SZ\_D & {DEC\_SZ{(reg\_addr == DMA2SZ )}});

1. 读写模式选择

reg\_write = |per\_we & reg\_sel;

reg\_read = ~|per\_we & reg\_sel;

1. 把reg\_dec分读写两种情况赋值给reg\_wr和reg\_rd

wire [DEC\_SZ-1:0] reg\_wr = reg\_dec & {DEC\_SZ{reg\_write}};

wire [DEC\_SZ-1:0] reg\_rd = reg\_dec & {DEC\_SZ{reg\_read}};

1. 把数据输入送入对应的寄存器里面

reg [15:0] dmactl0

wire dmactl0\_wr = reg\_wr[DMACTL0];

always @ (posedge mclk or posedge puc\_rst)

if (puc\_rst) dmactl0<= 16'h0000;

else if (dmactl0\_wr) dmactl0<= per\_din;