# LSU

LSU gồm 3 khối input\_buffer, output\_buffer và sram. Ta dùng decoder để xác định đại chỉ thanh ghi khi cần write vào output\_buffer để giao tiếp với ngoại vi hoặc là lưu giá trị vào sram. Ngõ ra là mux dùng để chọn tín hiệu ta cần load tương ứng.

A table with text and symbols

Description automatically generatedA diagram of a machine

Description automatically generatedCác thanh ghi được mapping như sau:

## Input buffer

Additional Components: flag\_buf

Input buffer dùng để lưu các giá trị tương ứng được gắn với switchs hoặc bottons vào hai register 32 bit tương ứng. A close-up of letters

Description automatically generated

Như yêu cầu từ milstone2, phân vùng địa chỉ cho Switches là từ 0x7800 đến 0x780F tương ứng với 15 đường địa chỉ mỗi địa chỉ đại điện cho 4 register 32 bit. Nhưng ta chỉ dùng 17 bit thấp của thanh ghi có địa chỉ là 0x7800 để nối vào các switch của DE2. Thực hiện tương tự với botton.

|  |  |
| --- | --- |
| Switchs | 0x7800 🡪 0x7803 |
| 0111 1000 00|00 00xx |
| Bottons | 0x7810 🡪 0x7813 |
| 0111 1000 00|01 00xx |
| Flag | 0x7820 🡪 0x7823 |
| 0111 1000 00|10 00xx |

A diagram of a rectangular object with text

Description automatically generatedThanh ghi flag là cờ trạng thái thông báo có cho phép ghi hay đọc tại 1 thời điểm hay không. Bit Flag[0] chính là tín hiệu o\_ACK được nới xung.

A graph with black lines

Description automatically generated

Từ bảng địa chỉ ở trên xác định ứng với từng bit địa chỉ nào các switchs hoặc btn hoặc flag được trỏ.

sw\_en = ~addr[15] &

(addr[14] & addr[13] & addr[12] & addr[11]) &

(~addr[10] & ~addr[9] & ~addr[8] & ~addr[7] & ~addr[6] & ~addr[5] & ~addr[4] & ~addr[3] & ~addr[2]);

btn\_en = ~addr[15] &

(addr[14] & addr[13] & addr[12] & addr[11]) &

(~addr[10] & ~addr[9] & ~addr[8] & ~addr[7] & ~addr[6] & ~addr[5] & addr[4] & ~addr[3] & ~addr[2]);

flag\_en = ~addr[15] &

(addr[14] & addr[13] & addr[12] & addr[11]) &

(~addr[10] & ~addr[9] & ~addr[8] & ~addr[7] & ~addr[6] & addr[5] & ~addr[4] & ~addr[3] & ~addr[2]);

Từ địa chỉ ngõ vào ta xác định được các thành phần nào đang được trỏ.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Flag\_en | btn\_en | sw\_en | Sel | Sel[1:0] |
| 0 | 0 | 1 | 0 | 00 |
| 0 | 1 | 0 | 1 | 01 |
| 1 | 0 | 0 | 2 | 10 |

Từ bảng xác định được sel để nối ra ngõ ra input\_buf từ đó đọc ngược giá trị về khi có lệnh load.

A diagram of a computer program

Description automatically generated

## Decode Stage

Khi có tín hiệu wren thì phát xung enable cho output\_buf hoặc sram

|  |  |
| --- | --- |
| Bank | Bank address |
| Input\_buf | 0x78XX |
| output\_buf | 0x70XX |
| Sram | 0x20XX 🡪 0x3FXX |

A diagram of a computer network

Description automatically generated Xác định logic cho output\_buf\_en và sram\_buf\_en

|  |  |  |  |
| --- | --- | --- | --- |
| Bank address | binary address[15:8] | Output\_buf\_en | Sram\_buf\_en |
| 0x70XX | 0111\_0000 | 1 | 0 |
| 0x20XX 🡪 0x3FXX | 001X\_XXXX | 0 | 1 |

output\_buff\_en = ~addr[15] & ( addr[14] & addr[13] & addr[12] ) &

~ ( addr[11] | addr[10] | addr[9] | addr[8] ) |

i\_lsu\_wren

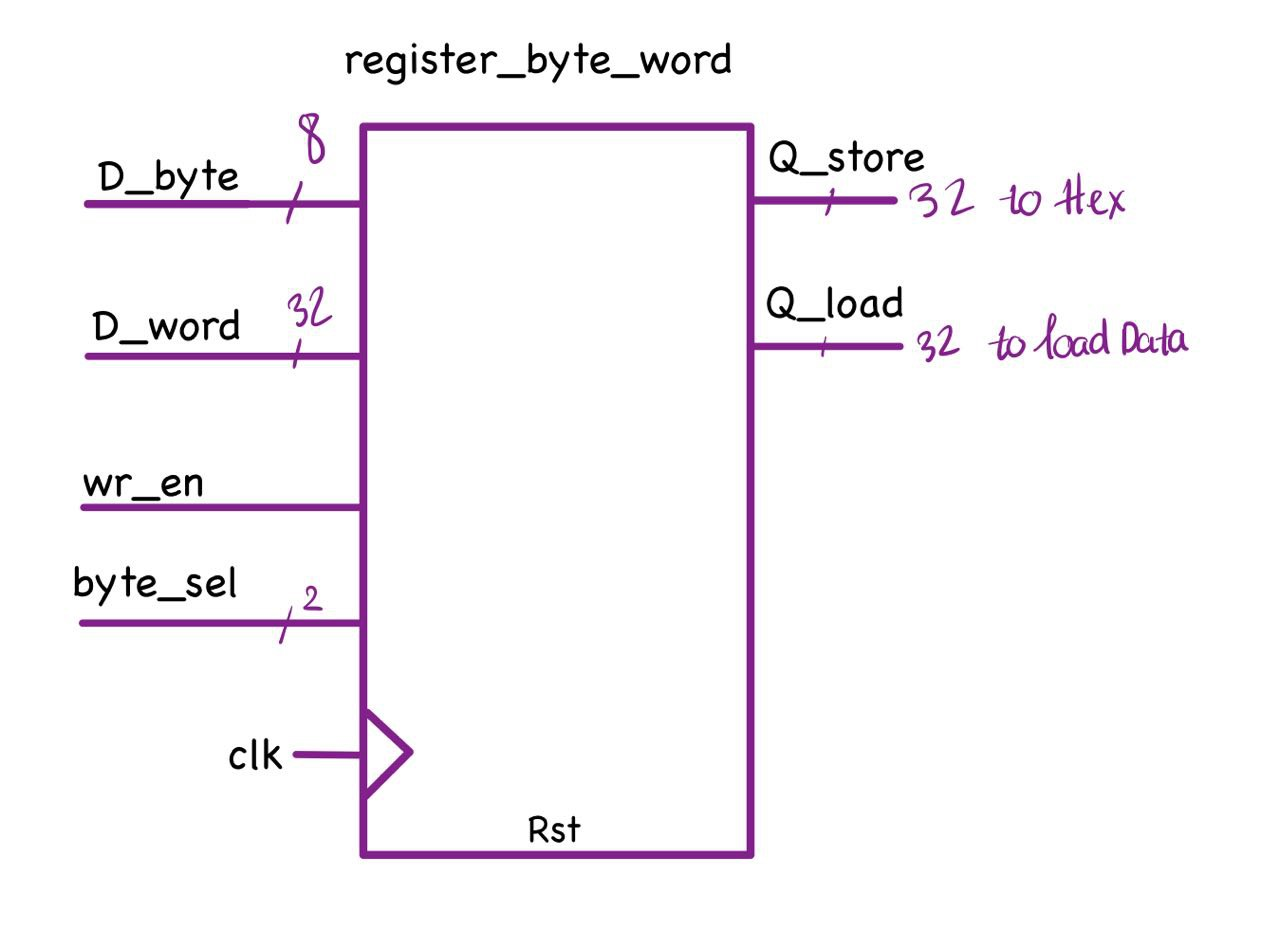
And với tín hiệu i\_lsu\_wren để khóa enable

Sram\_en = ~addr[15] & ~ addr[14] & addr[13] & i\_lsu\_wren

Tại một thời điểm chỉ có một thành phần được enable.

## Output buffer

### IP register\_byte\_word

Từ một register bình thường ta phát triển thêm thành IP register có khả năng load/store theo từng byte hoặc word ( 4 byte ). Khi Store theo word (sw) thì nạp thẳng 32 bit giá trị từ rs2\_data vào thanh ghi. Còn khi store theo byte (sb) thì nạp 8 bit thấp của rs2\_data vào thanh ghi theo opset được cộng. Ngoài ra còn có tín hiệu wb\_sel để còn định dạng theo byte hay theo word.

|  |  |
| --- | --- |
| wb\_sel | Định dạng thanh ghi |
| 1 | Theo word |
| 0 | Theo byte |

byte\_sel[1:0] dùng để chọn byte muốn nạp dữ liệu được assign với addr[1:0]

|  |  |  |
| --- | --- | --- |
| 0x702X | 0000 | Hex0🡪 Hex3 |
| 0001 |
| 0010 |
| 0011 |
| 0100 | Hex4🡪Hex7 |
| 0101 |
| 0110 |
| 0111 |
| 1000 | Không sử dụng |

Từ bảng ta thấy bit addr[3] dùng để phân biệt 2 bank Hex còn các byte phân biệt thông qua addr[1:0]

### register\_byte\_word schematic

A diagram of a block diagram

Description automatically generated

|  |  |
| --- | --- |
| **Pin** | **Tính năng** |
| D\_word [31:0] | Nạp từ rs2\_data |
| D\_byte [7:0] | 8 bit thấp của rs2\_data |
| Reg\_byte\_en | Enable từng thanh ghi tương ứng nếu lệnh theo dạng word thì enable cả 4 byte |
| Byte\_sel[1:0] | 2 bit LSB của addr để chọn byte |
| Q\_store[31:0] | Nối cứng chân với ngoại vi |
| Q\_load [31:0] | Để load ngược lại thanh ghi |

A white sheet with red writing on it

Description automatically generatedA diagram of a computer program

Description automatically generated

Figure Sơ đồ khối ouput\_buffer

Bus data được nối với tất cả các thanh ghi, muốn lưu vào thanh ghi nào phải trỏ đúng địa chỉ để decode stage cấp tín hiệu enable đúng cho các thanh ghi red, green... tương ứng. Ta có thể chọn nạp theo byte hoặc theo word theo như signal wb\_sel điều khiển.

## Output\_mux

Output để chọn từng bank input khi dùng lệnh load. Như phân vùng địa chỉ ở trên các ngõ vào ip\_en, op\_en, sram\_en, khi được trỏ đúng địa chỉ của nó

ip\_en = ~addr[15] &

(addr[14] & addr[13] & addr[12] & addr[11]) &

(~addr[10] & ~addr[9] & ~addr[8] & ~addr[7] & ~addr[6]);

op\_en = ~addr[15] &

(addr[14] & addr[13] & addr[12]) &

(~addr[11] & ~addr[10] & ~addr[9] & ~addr[8] & ~addr[7] & ~addr[6] );

sram\_en = ~addr[15] & ~addr[14] & addr[13];

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ip | op | sram | Sel | Sel[1:0] |
| 1 | 0 | 0 | 0 | 00 |
| 0 | 1 | 0 | 1 | 01 |
| 0 | 0 | 1 | 2 | 10 |

sel[1] = ~ip\_en & ~op\_en & sram\_en;

A diagram of a circuit board

Description automatically generatedsel[0] = ~ip\_en & op\_en & ~sram\_en;