|  |
| --- |
| CNN CONTROLLER  Design Document |
| Designed by S.J KO |
| · Contents  1. Properties / Information  2. FSM  3. Block design |
| 1 |

|  |
| --- |
| 1. Properties / Information  EMB00001820730e1) Design Overview  This image shows the overall communication. The signal is sent to Adrress Generator through AXI and each BRAM is made of true dual port to communicate with both Adrress Generator and CNN IP Controller. True dual port has two terminals are both read and writeable. Register communicates with CNN IP Controller with various parameter values. |
| 2 |
| 1. Properties / Information   EMB00001820730f2) Outline of CNN Controller  The purpose of this is to perform convolution operations using each BRAM and CNN IP Controller that already contain the data we need. When we perform this operation, we must consider Kernel size, Stride and Padding. Our input image size is 32x32. when padding operates, input image size is 34x34 and we can check padding location by conditional logic. Stride and kernel size will affect in Convolution State. |
| 3 |
| 1. Properties / Information  3) Condition  - Assumed ARM interface code and BRAM data are already exist.  - Using three slave registers and three block memories.  - Need to consider kernel size, stride, padding  Padding is used to prevent data loss during convolution operation.  ( You can receive images that have already been padded from the beginning as input )  Stride and kernel size affect the number and delay of existing  PE modules in the state of Conv. |
| 4 |
| 1.Properties / Information  4) Design plan (Kernel size, stride)  EMB000018207310  Kernel size defines the number of PE modules. if using 3\*3 filter, the number of filter is 9, using 5\*5 filter, the number of filter is 25.  EMB000018207311  We will use shift register for considering stride.   |  | | --- | | 5 |  |  |  |  |  | | --- | --- | --- | --- | | 1. Properties / Information  4) Design plan (Padding)  EMB000018207312  In C code, the condition of padding is ‘if( i<padding || j<padding || i>= image size + padding || j>= image size + padding )’ Assuming you already know the size of input image, if you have any i, j has any 0 or 34 when the padding is on, you can fill in the blanks. |  |  |  | | 6 |  |  |  | |

|  |
| --- |
| 2. FSM  1) FSM Overview  EMB000018207313  This FSM produced a done signal for the condition to move from each state to the next. When the start signal goes high level in initial state, it can move on to the next state of Read. As such, the conditional signal can only pass when it is high level. |
| 7 |
| 2. FSM  2) IDLE state / Write state  EMB000018207314EMB000018207315  The initial state of the FSM is IDLE. So, this is the initial state of waiting for the start signal and then moving on to the next state only when the signal is received. Likewise, the state of Write does the same. |
| 8 |
| 2. FSM  EMB0000182073163) Read state  When start signal is high level, Read state begins. In the Read state, the wea of input BRAM and weight BRAM are zero because the data stored in input BRAM and weight BRAM must be read. Since both values are required for convolution calculation, a done signal can be made for each BRAM and moved to the next state when input\_done signal and weight\_done signal are high. |
| 9 |
| 2. FSM  4) Conv state  EMB000018207317  When input\_done and weight\_done signal are high level, Conv state begins.  In the Conv state, input data read state is multiplied by the weight data and calculated using the PE module, which adds the previous value. If you choose 3\*3 filter, the number of PE modules is 9, and choose 5\*5 filter, the number of PE modules is 25. The number of delays depends on the value of stride |
| 10 |

|  |
| --- |
| 3. Block Diagram  1) Address Map Block diagram  EMB000018207318  EMB000018207319 |