A fazer:

1. colocar imagens do circuito
2. escrever o nome dos registos correctamente
3. explicar a comunicação com o computador
4. explicar porque escolhemos processar uma linha inteira em vez de 32 bits
5. rever o texto e pô-lo mais bonito

talvez mais coisas mas agora não me lembro

as linhas sublinhadas são aquelas em que a ideia está lá mas o português e a organização da frase não estão muito bem

Introdução

Este projeto tem como objetivo a criação de um circuito a ser implementado numa FPGA que faça algumas operações básicas de processamento de imagens. As imagens a serem processadas têm resolução máxima de 128x128 e são monocromáticas binárias (as únicas cores possíveis são branco ou preto). Pretende-se que este processamento seja o mais rápido possível, sem preocupação com eficiência energética.

Requisitos

O circuito deve ser capaz de realizar as seguintes operações morfológicas básicas sobre quaisquer imagens digitais binárias com dimensões de 128x128 pixéis:

* Erosão
* Dilatação
* Fecho
* Abertura
* Extração de Contornos

**Operações Simples**

Pretende-se projetar duas operações simples:

* Dilatação
* Erosão

Estas operações distinguem-se pelo valor do pixel que têm em consideração, a dilatação considera os pixéis com valor 1 e a erosão tem em conta os pixéis com valor 0. Cada uma destas operações percorre todos os pixéis da imagem original avaliando uma sub-imagem com dimensões 3x3 bits com centro no pixel a ser atual. No caso de uma dilatação, o pixel da imagem resultante correspondente ao centro da sub-imagem tem valor 1 caso um dos pixéis da sub-imagem tenha valor 1. Desta forma, temos o seguinte algoritmo para a operação de Dilatação:

**for** row **in** 0 **to** 127

**for** col **in** 0 **to** 127

**image(**row**,** col**)** **=** **image(**row**-**1**,** col**-**1**)** or **image(**row**-**1**,** col**)** or **image(**row**-**1**,** col**+**1**)**

or **image(**row**,** col**-**1**)** or **image(**row**,** col**)** or **image(**row**,** col**+**1**)**

or **image(**row**+**1**,** col**-**1**)** or **image(**row**+**1**,** col**)** or **image(**row**+**1**,** col**+**1**)**

**end** **for;**

**end** **for;**

Uma operação de erosão segue exatamente o mesmo algoritmo mas as portas lógicas OR utilizadas são substituídas por portas AND.

1. explicar porque escolhemos processar uma linha inteira em vez de 32 bits

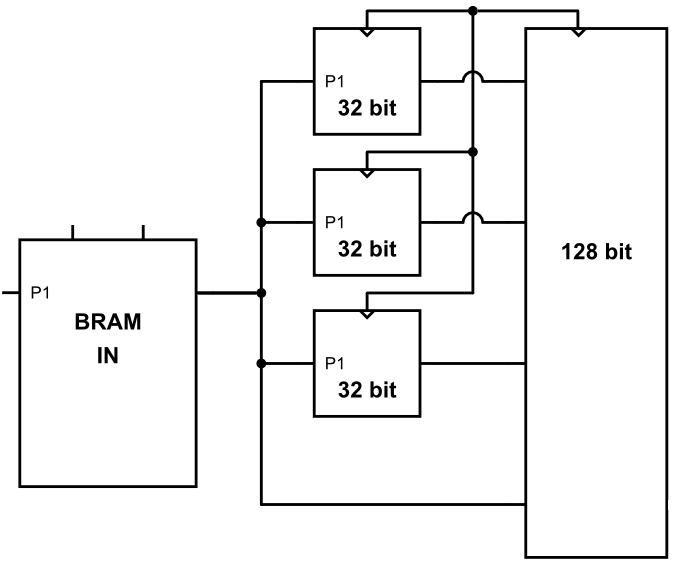
O circuito é composto por 3 partes distintas, a Unidade de Controlo, a *datapath* e as 2 memórias de leitura e escrita.

A Unidade de Controlo é responsável pela gestão de todos os recursos utilizados no circuito, define quais os registos ativados em cada ciclo, os sinais de seleção dos multiplexers e o endereçamento das memórias de leitura e de escrita. O endereçamento das memórias é feito através de um contador único. A saída do contador endereça a memória de leitura, para endereçar a memória de escrita são utilizados X *shift-registers* que produzem um atraso de X ciclos no endereço do contador. Abaixo encontra-se a seguinte máquina de estados utilizada na Unidade de Controlo:

1. Imagem da maquina de estados de nível 0

A *datapath* é responsável pelo processamento em si de cada pixel da imagem, sendo composta por 3 partes:

Na entrada da *datapath* tem-se um bloco de 4 registos utilizados para auxiliar a leitura de uma linha inteira da imagem original (128 bits), são lidos os primeiros 3 blocos de 32 bits de cada linha desde os mais significativos para os menos significativos e são armazenados pela mesma ordem nos registos de 32 bits R0, R1 e R2, após a leitura do 4º bloco de 32 bits este é armazenado nos 32 bits menos significativos do registo Rin e o conteúdo dos 3 registos anteriores é carregado, pela mesma ordem, par aos bits mais significativos do registo Rin. Esta arquitetura permite que sejam feitas leituras concorrentes da memória enquanto deixando a linha armazenada em Rin inalterada durante 3 ciclos de leitura.



Uma vez que para processar uma linha são necessárias: a linha acima e a linha abaixo da mesma é necessário ter a três linhas disponíveis em cada ciclo de processamento de uma linha. Para resolver este problema utilizou-se 3 registos de 128 bits ligados em série à saída do registo Rin: Ri-1, Ri, e Ri+1; estes são responsáveis por armazenar, respetivamente, as linhas anterior, atual e a linha a seguir à linha a ser processada. A cada atualização do registo Rin a nova linha lida é carregada para o registo Ri+1 e o conteúdo atual de cada um dos 3 registos é transferido para o registo seguinte. Desta forma, pode proceder-se ao processamento da linha contida em Ri com a garantia de que todas as suas dependências estão disponíveis. Durante a leitura da primeira linha da imagem os registos R-1, Ri e Ri+1 são carregados com o valor neutro da operação correspondente, no caso da dilatação são carregados com o valor 0 e no caso da erosão com o valor 1, de forma a tornar possível o processamento dos extremos da imagem.

1. Imagem com os registos de shift

O processamento, propriamente dito, é realizado pelo bloco de lógica à saída dos 3 registos anteriores. Tem-se 2 blocos com a mesma forma mas com portas lógicas contrárias, para se executar uma erosão são utilizadas portas lógicas AND e para uma dilatação são utilizadas portas lógicas OR. É necessário utilizar-se um multiplexer na saída deste bloco para selecionar qual dos dois resultados se pretende guardar no registo que armazena o resultado da operação.

1. imagem com o bloco de logica, o multiplexer de seleção da saida e o registo de resultado

O resultado armazenado no registo de saída é carregado para a memória de escrita em blocos de 32 bits começando pelos blocos mais significativos. Esta seleção é feita com recurso a um multiplexer de 4-1 cujos bits de seleção são dados pelos 2 bits menos significativos do sinal de endereçamento da memória de escrita no ciclo correspondente.

Abaixo encontra-se o circuito completo que permite executar as duas operações simples:

1. Imagem com o circuito sem a memoria adicional e sem xor

**Operações Compostas**

* Fecho
* Abertura

As operações de Fecho e Abertura são operações compostas das duas anteriores. Uma operação de Fecho executa uma Dilatação seguida de uma Erosão e uma Abertura executa as mesmas duas operações pela ordem inversa.

Para que seja possível executar duas operações simples é necessário que o resultado da primeira operação seja armazenado para posterior consulta durante a operação seguinte. Para esse efeito recorreu-se a uma memória adicional. Utiliza-se assim 3 memórias BRAM, a memória de leitura, a memória de escrita e a memória auxiliar, o que está dentro dos recursos disponíveis na FPGA utilizada no laboratório (basys2 / spartan 3e?).

Para controlar a ordem das operações e determinar a partir de qual das duas memórias, memória de leitura ou memória auxiliar, é lida a imagem a ser processada pelo dispositivo em cada ciclo de execução utilizou-se uma Unidade de Controlo adicional permitindo assim fazer uso do circuito anteriormente utilizado anteriormente nas operações simples.

Obteve-se então o seguinte circuito que permite tanto obter resultados das duas operações simples bem como das duas operações compostas.

1. Colocar imagem do circuito com as 3 memorias mas ainda sem a xor

**Extração de Contornos**

Esta operação resume-se a executar uma operação de Erosão sobre a imagem original e no final comparar esse resultado com a imagem original, extraindo a diferença entre as duas.

A diferença entre dois pixéis é dada pela função lógica XOR. Fazendo uso da mesma, obteve-se o seguinte circuito que permite obter a diferença entre cada bit da imagem original e da imagem resultante:

1. Imagem do circuito com foco apenas na parte da xor

De forma integrar o esquema anterior no circuito já implementado recorreu-se a um multiplexer que permite selecionar se o resultado que se pretende corresponde ao resultado de qualquer operação ou a diferença entre esse mesmo resultado e a imagem original. Para que seja executada uma operação de Extração de Contornos a operação simples a ser selecionada deverá ser uma Erosão e o sinal de seleção do multiplexer X deve estar ativado. Atenção que não é necessário recorrer à memória auxiliar para executar esta operação, logo pode ser executa em apenas um ciclo de execução.

Abaixo está demonstrado o circuito final com todas as operações implementadas:

1. Imagem do circuito final

**Comunicação Entre Computador e Dispositivo**

As imagens que se pretende processar são carregadas para o dispositivo através do porto de escrita A da memória de leitura e após serem processadas são exportadas pelo porto de leitura A da memória de escrita. Ambos estes portos são portos de 8 bits o que significa que são escritos 8 bits de cada vez na memória de leitura e lidos 8 bits de cada vez da memória de escrita. Por outro lado a comunicação entre o dispositivo e as memórias é feito por portos de 32 bits …………

3) explicar a comunicação com o computador