Pretende-se projetar um circuito capaz de realizar as seguintes operações morfológicas básicas sobre quaisquer imagens digitais binárias com dimensões de 128x128 bits:

* 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.

Abaixo encontra-se o circuito utilizado para executar estas operações:

Imagem com o circuito sem a memoria adicional

Imagem da maquina de estados de nível 0

É preciso adicionar aqui uma explicação da arquitetura do circuito

**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.

Uma vez que se pretende executar duas operações simples isto significa que o resultado da primeira operação deve ser armazenado e consultado na operação seguinte, de forma a obtermos o resultado final da respetiva operação composta. Para esse efeito recorreu-se a uma memória adicional.

Para controlar a ordem das operações e determinar de qual das duas memórias, memória de leitura ou memória auxiliar, é lida a imagem em cada operação 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.

**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, obtém-se o seguinte circuito que calcula a diferenças entre a imagem original e o resultado da operação de Erosão:

Imagem do circuito com foco apenas na parte da xor

Para integrar este esquema no circuito anterior recorre-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. Abaixo está demonstrado o circuito final com todas as operações implementadas:

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 …………