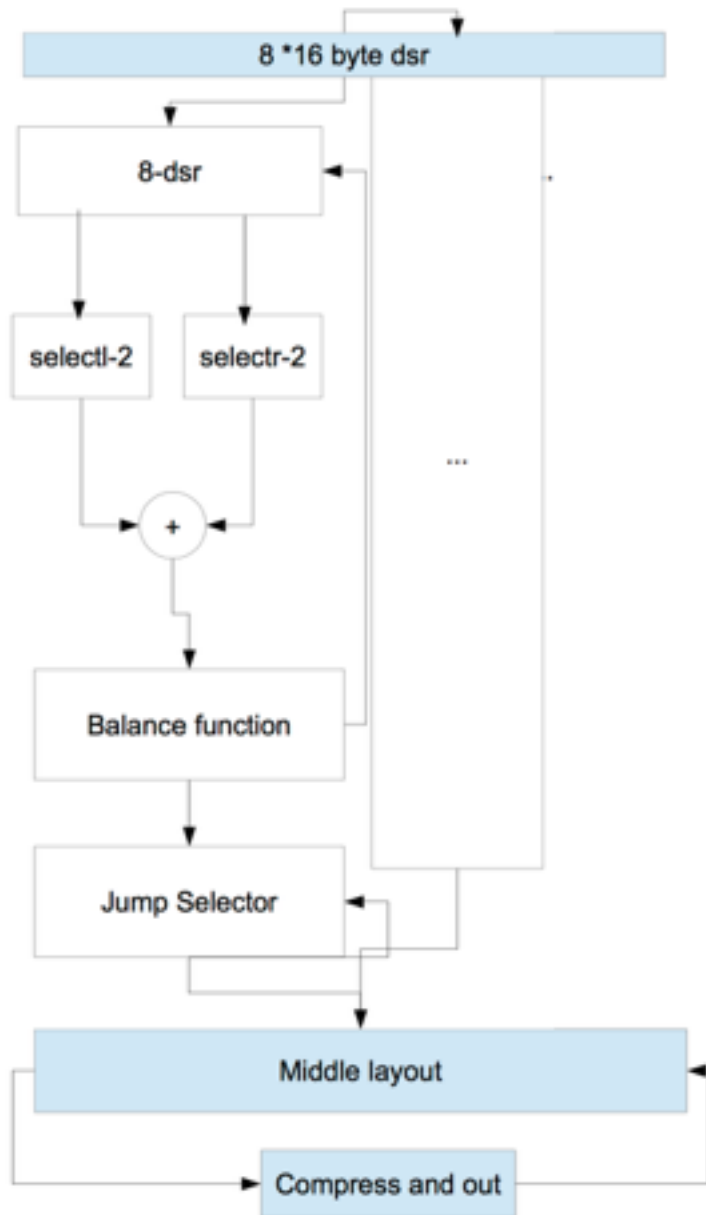


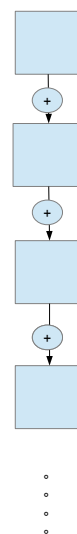
Introduction

略过无聊的介绍

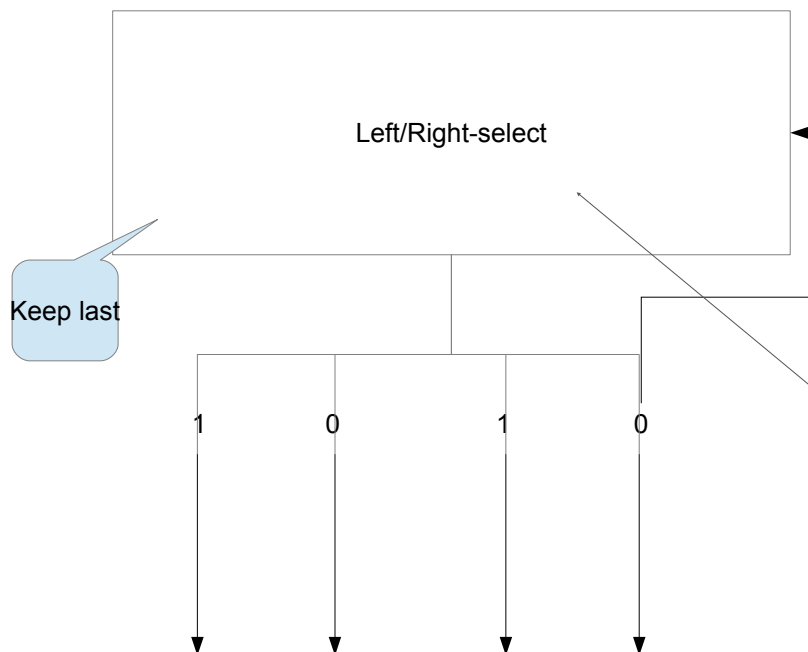
改进



- 用了KG的基本结构，在开始的128 DSR 后，分为 16X8 的字节流，然后再分别做DSR
- 得到的结果，拆成左右两半，亦或后，判断其01的平衡性.如果01差距过大，再进行 8-DSR。
- 8-DSR 受平衡函数的反馈影响。
- 加入了中间层，作为所有分支的汇总，其中会进行链式亦或得到
- 改进主要为了编程实现方便



中间层的构造比较简单。通过链式的或 和链接，得到4字节的字节流，最后通过压缩函数得到最后的结果
这之间的大量的记忆因子会用在下一个 字节的运算中

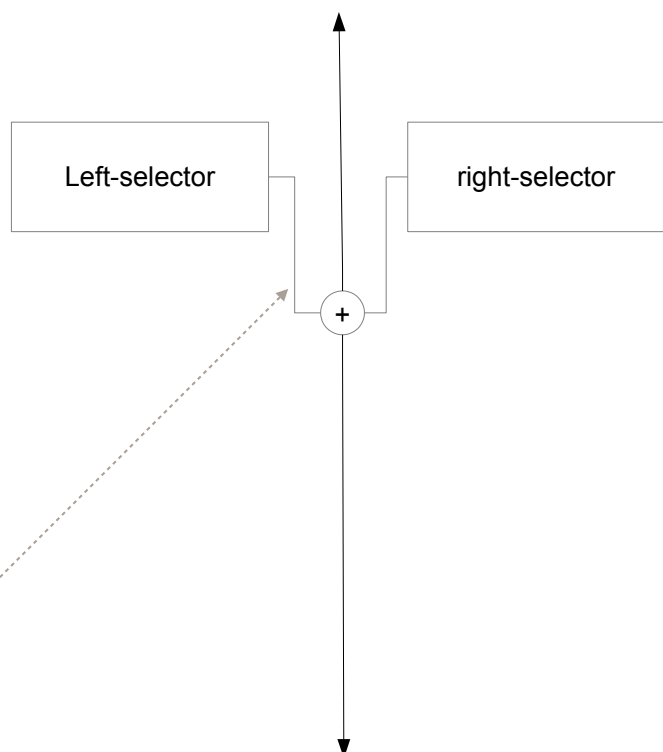


单个的左选择器如右，保留每次的输出
的最后一位作为反馈来影响新的的
输出

$\text{left} \ll 1 \wedge 0\text{xff};$
 $\text{right} \gg 1 \wedge 0\text{xff};$

if temp == old :

goto 8-DSR



bytes/out	
128-DSR	128
8-DSR	8
4-selector	4
B	4
Jump	2
Middle	4

字节输出的长度表