(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 106021171 A
(43)申请公布日 2016.10.12

(21)申请号 201610299347.X

(22)申请日 2016.05.06

(71)申请人 东南大学—无锡集成电路技术研究所
地址 214135 江苏省无锡市新区菱湖大道99号

(72)发明人 杨锦江 闵婧 申艾麟 尹玲 李兆奇 明畅 葛伟

(74)专利代理机构 南京瑞弘专利商标事务所（普通合伙）32249
代理人 陈国强

(51)Int.Cl.
G06F 15/78(2006.01)
G06F 21/72(2013.01)

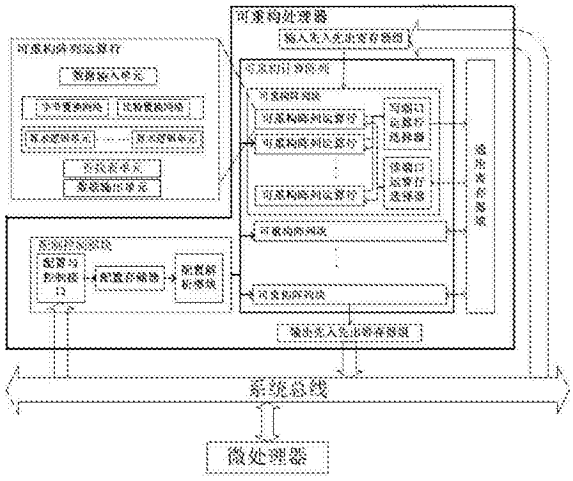
权利要求书3页 说明书6页 附图4页

(54)发明名称

一种基于大规模粗粒度可重构处理器的SM4-128的密钥扩展实现方法及系统

(57)摘要

本发明公开了一种基于大规模粗粒度可重构处理器的SM4-128的密钥扩展实现方法及系统，该系统包括可重构处理器、微处理器、系统总线；所述可重构计算阵列包括可重构计算阵列块，可重构计算阵列块包括可重构阵列运算行、写端口运算行选择器、读端口运算行选择器；所述微处理器通过系统总线分别与配置控制模块的配置与控制接口，可重构处理器的输入先入先出寄存器组连接，所述输入先入先出寄存器组连接可重构计算阵列，可重构计算阵列连接输出端连接可重构处理器，输出端连接可重构处理器通过系统总线与微处理器连接。本发明针对SM4-128密钥扩展方法，通过将多轮迭代在可重构处理器中部分展开和中间结果数据缓存的方式进行优化和加速。



CN 106021171 A

1.一种基于大规模粗粒度可重构处理器的SM4-128的密钥扩展系统，其特征在于：包括可重构处理器、微处理器、系统总线；

其中，所述可重构处理器包括配置控制模块、输入先入先出寄存器组、输出先入先出寄存器组、通用寄存器堆、可重构计算阵列；

所述配置控制模块包括依次连接的配置与控制接口、配置存储器、配置解析模块，配置控制模块的输出端连接可重构处理器；

所述可重构计算阵列包括可重构计算阵列块，可重构计算阵列块包括可重构阵列运算行、写端口运算行选择器、读端口运算行选择器；所述可重构阵列运算行的输出端连接写端口运算行选择器的输入端，写端口运算行选择器的输出端连接通用寄存器堆；所述读端口运算行选择器的输入端接入通用寄存器堆，读端口运算行选择器的输出端连接可重构阵列运算行；

其中，所述可重构阵列运算行包括算术逻辑单元、查找表单元、比特置换网络、字节置换网络以及数据输入单元和数据输出单元；

所述微处理器通过系统总线分别与配置控制模块的配置与控制接口，可重构处理器的输入先入先出寄存器组连接，所述输入先入先出寄存器组连接可重构计算阵列，可重构计算阵列连接输出端连接可重构处理器，输出端连接可重构处理器通过系统总线与微处理器连接；

其中，通过分析SM4-128密钥扩展的特征来确定SM4-128密钥扩展的运算流程，将多轮的SM4-128密钥扩展运算展开成一幅数据流图映射到可重构处理器中，通过多幅数据流图最终完成SM4-128密钥扩展的整个运算；

微处理器通过系统总线发送明文数据给可重构处理器，可重构处理器将明文数据存入输入先入先出寄存器组，并在最终计算完成后输出密文数据到输出先入先出寄存器组，并发送中断信号，最终由微处理器读出输出至输出先入先出寄存器组中的数据。

2.如权利要求1所述的基于大规模粗粒度可重构处理器的SM4-128的密钥扩展系统，其特征在于：所述的可重构处理器有M个可重构计算阵列块、1个通用寄存器堆、1个输入先入先出寄存器组和1个输出先入先出寄存器组，其中M取整数；其中M个可重构计算阵列块通过一个1个通用寄存器堆互相进行数据的储存、读取和传递；且多个可重构计算阵列块中相邻的两个可重构计算阵列块通过数据输入单元和数据输出单元连接；第一个可重构计算阵列块通过第一个可重构阵列运算行的数据输入单元与输入先入先出寄存器组相连，同时第M个可重构计算阵列块通过最后一个可重构阵列运算行的数据输出单元与输出先入先出寄存器组相连。

3.如权利要求2所述的基于大规模粗粒度可重构处理器的SM4-128的密钥扩展系统，其特征在于：所述的每个可重构计算阵列块包括N个可重构阵列运算行和1个读端口运算行选择器和1个写端口运算行选择器，其中N取整数；其每N个可重构阵列运算行共享1个通用寄存器堆的读端口和写端口；在SM4-128密钥扩展运算中可重构阵列运算行通过通用寄存器堆读出各种缓冲数据如CK[i]，K[i]以及各种临时的消息摘要，其中CK[i]为32比特的固定参数，K[i]为32比特的扩展密钥，同时向通用寄存器堆写入消息摘要的每轮计算的中间值以及缓冲数据CK[i]及K[i]，这些缓冲数据被其他可重构计算阵列读出用于下一轮计算。

4.如权利要求3所述的基于大规模粗粒度可重构处理器的SM4-128的密钥扩展系统，其

特征在于:所述的可重构阵列运算行包括X_1个数据输入单元,X_2个数据输出单元,X_3个字节置换网络,X_4个比特置换网络和X_5个8位算术逻辑单元,X_6个查找表单元,其中X_1,X_2,X_3,X_4,X_5和X_6均取整数;数据经过数据输入单元,由选择器通过读取并解析不同的配置信息来选择数据流入的字节置换网络和比特置换网络;字节置换网络与比特置换网络的输出分为X_5个8位的数据分别固定对应于X_5个8位算术逻辑单元,并行运算$X_5/4$组SM4-128密钥扩展数据;每个算术逻辑单元使用数据选择器选择任意三个置换网络的输出作为其输入;数据输出单元暂存算术逻辑单元的结果并读取配置信息决定将数据输出到先入先出寄存器组、下一个可重构阵列运算行或通用寄存器堆。

5.如权利要求4所述的基于大规模粗粒度可重构处理器的SM4-128的密钥扩展系统,其特征在于:所述算术逻辑单元及显示查找表可实现异或运算、与运算、直通输出、查表操作等运算操作;同时每个算术逻辑单元有最多3个输入和最多2个输出,其中算术逻辑单元执行上述运算操作的同时,支持任选一个输入作为输出;每4个8位的算术逻辑单元通过进位端口连接成为1个32位的算术逻辑单元;每4个可重构阵列运算行共享一个显示查找表,来实现查表操作。

6.如权利要求1-5任一所述的基于大规模粗粒度可重构处理器的SM4-128的密钥扩展系统,其特征在于:该系统的密钥扩展流程包括如下6个步骤,对于32个32比特的扩展密钥,对(1)中的操作执行一次后,对(2)～(6)步骤顺序操作并循环32次,即可得到rk[i](其中0≤i≤31):

(1)密钥初始化:通过加密密钥MK[i]及系统参数FK[i]进行异或操作得到K[i],(其中0≤i≤3).对每个i值,MK[i]及对应FK[i]进行异或操作,得到K[i]。每32比特的MK[i]与FK[i]在算术逻辑单元中执行MK[i]+FK[i]的异或操作,并将计算结果存入通用寄存器中;

(2)生成m[i]:数据输入单元将K[i+1],K[i+2],K[i+3]载入可重构运算单元行中,在经过字节置换网络进行移位后,再对K[i+1],K[i+2],K[i+3]三者进行异或操作,将结果m[i]输出至缓存单元中;

(3)生成t[i]:数据输入单元将CK[i]及缓存单元中的数据m[i]载入可重构运算单元行中,在经过字节置换网络后,在逻辑运算单元中对CK[i]及m[i]进行异或操作,将结果t[i]存入缓存单元中;

(4)查表操作:数据输入单元将t[i]从缓存单元中载入至可重构运算单元行中。t[i]在通过字节置换网络后,在显示查找表中对t[i]进行查表操作,得到B[i],并将B[i]存入缓存单元中;

(5)线性变换:数据输入单元将B[i]从缓存单元中载入至可重构阵列单元行中。B[i]在经过比特置换网络后,生成中间数据B1[i],B2[i],将B1[i],B2[i],B[i]三者进行异或操作,生成T[i],并将T[i]存入缓存单元;

(6)生成rk[i]:数据输入单元将T[i]从缓存单元中载入至可重构阵列单元行中,对T[i]及K[i]进行异或操作,得到K[i+4],即rk[i]。

7.一种基于大规模粗粒度可重构处理器的SM4-128密钥扩展实现方法,其特征在于:包括以下步骤:

(1)分析SM4-128密钥扩展的计算特点,并归纳出数据流图;

(2)确定数据流图之后,针对可重构处理器的硬件特点,在了解其各寄存器、运算器以

及各功能模块的作用机制的情况下配置可重构处理器，并生成配置信息；

(3)通过微处理器将配置信息以及所需要的各种初始数据存入相应的存储器中；

(4)最后微处理器启动可重构处理器，并将配置信息及数据发送给可重构处理器；

(5)当可重构处理器完成当前任务后，发送中断信号。

一种基于大规模粗粒度可重构处理器的SM4-128的密钥扩展实现方法及系统

技术领域

[0001] 本发明涉及嵌入式可重构系统领域，尤其涉及一种应用于通信、加密等领域的基于大规模粗粒度嵌入式可重构系统及其处理方法。

背景技术

[0002] 通用处理器与专用集成电路(ASIC)是传统的计算机系统结构领域的两大主流方法。然而，随着应用领域对系统的性能、能耗、上市时间等指标需求的不断提高，这两种传统计算模式的弊端就暴露出来。

[0003] 通用处理器方法适用范围广，但是计算效率低，专用集成电路虽然可以提高计算速度和计算效率，满足性能需求，但是ASIC器件的灵活性很差。

[0004] 为了在灵活性和计算效率之间实现很好的权衡，可重构计算(reconfigurable computing)技术应运而生。可重构计算是当前计算机系统结构领域的发展趋势之一，它的架构介于通用处理器和ASIC之间，并且综合了二者长处。它通过对可重构设备进行配置，可以使之由一个通用的计算平台转化为一个专用的硬件系统，以完成具体的计算任务，相当于计算任务同时在时间和空间上展开，显示出了应用的灵活性和很高的计算性能。此外，可重构计算技术还具有系统能耗低、可靠性高、上市时间短等优势。这些优势使得可重构计算技术在各个应用领域尤其是嵌入式应用领域有着广阔的应用前景。很多在嵌入式领域中的主流应用，例如多媒体应用、加/解密应用以及通信应用等都非常适合利用可重构计算技术实现。当前的可重构计算技术主要还是用于尖端技术领域中的计算平台，但随着可重构逻辑器件成本逐渐降低，运行时可重构计算技术不断完善，我们有理由相信可重构计算技术具备的种种优势会使其在更多的领域里大有作为。

[0005] 目前国内外已研究有多重可重构系统，如ReMAP，AsAP，DRP等。但是，这些阵列的互联方式较为简单，在SM4-128方法的运算中需要大量的中间数据存储以及较多的轮数，因此运算的效率和速度较低。传统的可重构计算系统在SM4-128的运算效率与运算周期方面存在较大问题。

发明内容

[0006] 为了克服现有技术中存在的不足，本发明的目的是提供一种基于大规模粗粒度可重构处理器的SM4-128的密钥扩展实现方法及系统，利用可重构技术的并行性处理、运算模块独立可配置等优点，在支持一定的灵活性的同时，通过提高对SM4-128密钥扩展的并行度以及优化流水线等方法以实现SM4-128密钥扩展的高效运算。

[0007] 为实现上述目的，本发明采用的技术方案为：

[0008] 一种基于大规模粗粒度可重构处理器的SM4-128的密钥扩展系统，包括可重构处理器、微处理器、系统总线；

[0009] 其中，所述可重构处理器包括配置控制模块、输入先入先出寄存器组、输出先入先

出寄存器组、通用寄存器堆、可重构计算阵列；

[0010] 所述配置控制模块包括依次连接的配置与控制接口、配置存储器、配置解析模块，配置控制模块的输出端连接可重构处理器；

[0011] 所述可重构计算阵列包括可重构计算阵列块，可重构计算阵列块包括可重构阵列运算行、写端口运算行选择器、读端口运算行选择器；所述可重构阵列运算行的输出端连接写端口运算行选择器的输入端，写端口运算行选择器的输出端连接通用寄存器堆；所述读端口运算行选择器的输入端接入通用寄存器堆，读端口运算行选择器的输出端连接可重构阵列运算行；

[0012] 其中，所述可重构阵列运算行包括算术逻辑单元、查找表单元、比特置换网络、字节置换网络以及数据输入单元和数据输出单元；

[0013] 所述微处理器通过系统总线分别与配置控制模块的配置与控制接口，可重构处理器的输入先入先出寄存器组连接，所述输入先入先出寄存器组连接可重构计算阵列，可重构计算阵列连接输出端连接可重构处理器，输出端连接可重构处理器通过系统总线与微处理器连接；

[0014] 其中，通过分析SM4-128密钥扩展的特征来确定SM4-128密钥扩展的运算流程，将多轮的SM4-128密钥扩展运算展开成一幅数据流图映射到可重构处理器中，通过多幅数据流图最终完成SM4-128密钥扩展的整个运算；

[0015] 微处理器通过系统总线发送明文数据给可重构处理器，可重构处理器将明文数据存入输入先入先出寄存器组，并在最终计算完成后输出密文数据到输出先入先出寄存器组，并发送中断信号，最终由微处理器读出输出至输出先入先出寄存器组中的数据。

[0016] 首先对配置控制模块中的配置存储器进行初始化，微处理器将所需要的配置信息通过配置与控制接口发送到配置存储器中，然后通过配置解析模块解析配置存储器，实现对计算阵列的配置、启动以及切换操作。

[0017] 进一步的，所述的可重构处理器有M个可重构计算阵列块、1个通用寄存器堆、1个输入先入先出寄存器组和1个输出先入先出寄存器组，其中M取整数；其中M个可重构计算阵列块通过一个1个通用寄存器堆互相进行数据的储存、读取和传递；且多个可重构计算阵列块中相邻的两个可重构计算阵列块通过数据输入单元和数据输出单元连接；第一个可重构计算阵列块通过第一个可重构阵列运算行的数据输入单元与输入先入先出寄存器组相连，同时第M个可重构计算阵列块通过最后一个可重构阵列运算行的数据输出单元与输出先入先出寄存器组相连。作为优选方案，可重构处理器有10个可重构计算阵列块.

[0018] 进一步的，所述的每个可重构计算阵列块包括N个可重构阵列运算行和1个读端口运算行选择器和1个写端口运算行选择器，其中N取整数；其每N个可重构阵列运算行共享1个通用寄存器堆的读端口和写端口；在SM4-128密钥扩展运算中可重构阵列运算行通过通用寄存器堆读出各种缓冲数据如CK[i],K[i]以及各种临时的消息摘要，其中CK[i]为32比特的固定参数，K[i]为32比特的扩展密钥，同时向通用寄存器堆写入消息摘要的每轮计算的中间值以及缓冲数据CK[i]及K[i]，这些缓冲数据被其他可重构计算阵列读出用于下一轮计算。作为优选方案，可重构阵列块包括4个可重构阵列运算行。

[0019] 进一步的，所述的可重构阵列运算行包括X_1个数据输入单元，X_2个数据输出单元，X_3个字节置换网络，X_4个比特置换网络和X_5个8位算术逻辑单元，X_6个查找表单元，其中X_1，

X_2,X_3,X_4,X_5和X_6均取整数;数据经过数据输入单元,由选择器通过读取并解析不同的配置信息来选择数据流入的字节置换网络和比特置换网络;字节置换网络与比特置换网络的输出分为X_5个8位的数据分别固定对应于X_5个8位算术逻辑单元,并行运算$X_5/4$组SM4-128密钥扩展数据;每个算术逻辑单元使用数据选择器选择任意三个置换网络的输出作为其输入;数据输出单元暂存算术逻辑单元的结果并读取配置信息决定将数据输出到先入先出寄存器组、下一个可重构阵列运算行或通用寄存器堆。

[0020] 进一步的,所述算术逻辑单元及显示查找表可实现异或运算、与运算、直通输出、查表操作等运算操作;同时每个算术逻辑单元有最多3个输入和最多2个输出,其中算术逻辑单元执行上述运算操作的同时,支持任选一个输入作为输出;每4个8位的算术逻辑单元通过进位端口连接成为1个32位的算术逻辑单元;每4个可重构阵列运算行共享一个显示查找表,来实现查表操作。

[0021] 进一步的,该系统的密钥扩展流程包括如下6个步骤,对于32个32比特的扩展密钥,对(1)中的操作执行一次后,对(2)～(6)步骤顺序操作并循环32次,即可得到rk[i](其中0≤i≤31):

[0022] (1)密钥初始化:通过加密密钥MK[i]及系统参数FK[i]进行异或操作得到K[i],(其中0≤i≤3).对每个i值,MK[i]及对应FK[i]进行异或操作,得到K[i]。每32比特的MK[i]与FK[i]在算术逻辑单元中执行MK[i]+FK[i]的异或操作,并将计算结果存入通用寄存器中;

[0023] (2)生成m[i]:数据输入单元将K[i+1],K[i+2],K[i+3]载入可重构运算单元行中,在经过字节置换网络进行移位后,再对K[i+1],K[i+2],K[i+3]三者进行异或操作,将结果m[i]输出至缓存单元中;

[0024] (3)生成t[i]:数据输入单元将CK[i]及缓存单元中的数据m[i]载入可重构运算单元行中,在经过字节置换网络后,在逻辑运算单元中对CK[i]及m[i]进行异或操作,将结果t[i]存入缓存单元中;

[0025] (4)查表操作:数据输入单元将t[i]从缓存单元中载入至可重构运算单元行中。t[i]在通过字节置换网络后,在显示查找表中对t[i]进行查表操作,得到B[i],并将B[i]存入缓存单元中;

[0026] (5)线性变换:数据输入单元将B[i]从缓存单元中载入至可重构阵列单元行中。B[i]在经过比特置换网络后,生成中间数据B1[i],B2[i],将B1[i],B2[i],B[i]三者进行异或操作,生成T[i],并将T[i]存入缓存单元;

[0027] (6)生成rk[i]:数据输入单元将T[i]从缓存单元中载入至可重构阵列单元行中,对T[i]及K[i]进行异或操作,得到K[i+4],即rk[i]。

[0028] 一种基于大规模粗粒度可重构处理器的SM4-128密钥扩展实现方法,包括以下步骤:

[0029] (1)分析SM4-128密钥扩展的计算特点,并归纳出数据流图;

[0030] (2)确定数据流图之后,针对可重构处理器的硬件特点,在了解其各寄存器、运算器以及各功能模块的作用机制的情况下配置可重构处理器,并生成配置信息;

[0031] (3)通过微处理器将配置信息以及所需要的各种初始数据存入相应的存储器中;

[0032] (4)最后微处理器启动可重构处理器,并将配置信息及数据发送给可重构处理器;

[0033] (5)当可重构处理器完成当前任务后,发送中断信号。

[0034] 有益效果:本发明基于大规模粗粒度动态可重构处理器,通过10个可重构阵列块包含多个运算单元,借助通用寄存器堆提高SM4-128算法的运算并行度,在具有一定灵活性的同时,提高SM4-128方法的运算效率,尽可能的减少运算周期。

附图说明

[0035] 图1为本发明的基于大规模粗粒度嵌入式可重构系统处理器框图;

[0036] 图2a-图2d为一个可重构阵列块的运算流图,共同构成本发明中SM4-128密钥扩展方法运算流图。

具体实施方式

[0037] 下面结合附图对本发明作更进一步的说明。

[0038] 如图1所示为一种基于大规模粗粒度可重构处理器的SM4-128的密钥扩展系统,包括可重构处理器、微处理器、系统总线;

[0039] 其中,所述可重构处理器包括配置控制模块、输入先入先出寄存器组、输出先入先出寄存器组、通用寄存器堆、可重构计算阵列;

[0040] 所述配置控制模块包括依次连接的配置与控制接口、配置存储器、配置解析模块,配置控制模块的输出端连接可重构处理器;

[0041] 所述可重构计算阵列包括可重构计算阵列块,可重构计算阵列块包括可重构阵列运算行、写端口运算行选择器、读端口运算行选择器;所述可重构阵列运算行的输出端连接写端口运算行选择器的输入端,写端口运算行选择器的输出端连接通用寄存器堆;所述读端口运算行选择器的输入端接入通用寄存器堆,读端口运算行选择器的输出端连接可重构阵列运算行;

[0042] 其中,所述可重构阵列运算行包括算术逻辑单元、查找表单元、比特置换网络、字节置换网络以及数据输入单元和数据输出单元;

[0043] 所述微处理器通过系统总线分别与配置控制模块的配置与控制接口,可重构处理器的输入先入先出寄存器组连接,所述输入先入先出寄存器组连接可重构计算阵列,可重构计算阵列连接输出端连接可重构处理器,输出端连接可重构处理器通过系统总线与微处理器连接;

[0044] 其中,通过分析SM4-128密钥扩展的特征来确定SM4-128密钥扩展的运算流程,将多轮的SM4-128密钥扩展运算展开成一幅数据流图映射到可重构处理器中,通过多幅数据流图最终完成SM4-128密钥扩展的整个运算;

[0045] 微处理器通过系统总线发送明文数据给可重构处理器,可重构处理器将明文数据存入输入先入先出寄存器组,并在最终计算完成后输出密文数据到输出先入先出寄存器组,并发送中断信号,最终由微处理器读出输出至输出先入先出寄存器组中的数据。

[0046] 首先对配置控制模块中的配置存储器进行初始化,微处理器将所需要的配置信息通过配置与控制接口发送到配置存储器中,然后通过配置解析模块解析配置存储器,实现对计算阵列的配置、启动以及切换操作。

[0047] 可重构处理器有M个可重构计算阵列块、1个通用寄存器堆、1个输入先入先出寄存

器组和1个输出先入先出寄存器组，其中M取整数；其中M个可重构计算阵列块通过一个1个通用寄存器堆互相进行数据的储存、读取和传递；且多个可重构计算阵列块中相邻的两个可重构计算阵列块通过数据输入单元和数据输出单元连接；第一个可重构计算阵列块通过第一个可重构阵列运算行的数据输入单元与输入先入先出寄存器组相连，同时第M个可重构计算阵列块通过最后一个可重构阵列运算行的数据输出单元与输出先入先出寄存器组相连。作为优选方案，可重构处理器有10个可重构计算阵列块.

[0048] 每个可重构计算阵列块包括N个可重构阵列运算行和1个读端口运算行选择器和1个写端口运算行选择器，其中N取整数；其每N个可重构阵列运算行共享1个通用寄存器堆的读端口和写端口；在SM4-128密钥扩展运算中可重构阵列运算行通过通用寄存器堆读出各种缓冲数据如CK[i],K[i]以及各种临时的消息摘要，其中CK[i]为32比特的固定参数，K[i]为32比特的扩展密钥，同时向通用寄存器堆写入消息摘要的每轮计算的中间值以及缓冲数据CK[i]及K[i]，这些缓冲数据被其他可重构计算阵列读出用于下一轮计算。作为优选方案，可重构阵列块包括4个可重构阵列运算行。

[0049] 可重构阵列运算行包括X_1个数据输入单元，X_2个数据输出单元，X_3个字节置换网络，X_4个比特置换网络和X_5个8位算术逻辑单元，X_6个查找表单元，其中X_1,X_2,X_3,X_4,X_5和X_6均取整数；数据经过数据输入单元，由选择器通过读取并解析不同的配置信息来选择数据流入的字节置换网络和比特置换网络；字节置换网络与比特置换网络的输出分为X_5个8位的数据分别固定对应于X_5个8位算术逻辑单元，并行运算$X_5/4$组SM4-128密钥扩展数据；每个算术逻辑单元使用数据选择器选择任意三个置换网络的输出作为其输入；数据输出单元暂存算术逻辑单元的结果并读取配置信息决定将数据输出到先入先出寄存器组、下一个可重构阵列运算行或通用寄存器堆。

[0050] 算术逻辑单元及显示查找表可实现异或运算、与运算、直通输出、查表操作等运算操作；同时每个算术逻辑单元有最多3个输入和最多2个输出，其中算术逻辑单元执行上述运算操作的同时，支持任选一个输入作为输出；每4个8位的算术逻辑单元通过进位端口连接成为1个32位的算术逻辑单元；每4个可重构阵列运算行共享一个显示查找表，来实现查表操作。

[0051] 该系统的密钥扩展流程包括如下6个步骤对于32个32比特的扩展密钥，对(1)中的操作执行一次后，对(2)～(6)步骤顺序操作并循环32次，即可得到rk[i](其中0≤i≤31)：

[0052] (1)密钥初始化：通过加密密钥MK[i]及系统参数FK[i]进行异或操作得到K[i],(其中0≤i≤3).对每个i值，MK[i]及对应FK[i]进行异或操作，得到K[i]。每32比特的MK[i]与FK[i]在算术逻辑单元中执行MK[i]+FK[i]的异或操作，并将计算结果存入通用寄存器中；

[0053] (2)生成m[i]：数据输入单元将K[i+1],K[i+2],K[i+3]载入可重构运算单元行中，在经过字节置换网络进行移位后，再对K[i+1],K[i+2],K[i+3]三者进行异或操作，将结果m[i]输出至缓存单元中；

[0054] (3)生成t[i]：数据输入单元将CK[i]及缓存单元中的数据m[i]载入可重构运算单元行中，在经过字节置换网络后，在逻辑运算单元中对CK[i]及m[i]进行异或操作，将结果t[i]存入缓存单元中；

[0055] (4)查表操作：数据输入单元将t[i]从缓存单元中载入至可重构运算单元行中。t

[i]在通过字节置换网络后，在显示查找表中对t[i]进行查表操作，得到B[i]，并将B[i]存入缓存单元中；

[0056] (5)线性变换：数据输入单元将B[i]从缓存单元中载入至可重构阵列单元行中。B[i]在经过比特置换网络后，生成中间数据B1[i]，B2[i]，将B1[i]，B2[i]，B[i]三者进行异或操作，生成T[i]，并将T[i]存入缓存单元；

[0057] (6)生成rk[i]：数据输入单元将T[i]从缓存单元中载入至可重构阵列单元行中，对T[i]及K[i]进行异或操作，得到K[i+4]，即rk[i]。

[0058] 一种基于大规模粗粒度可重构处理器的SM4-128密钥扩展实现方法，包括以下步骤：

[0059] (1)分析SM4-128密钥扩展的计算特点，并归纳出数据流图；

[0060] (2)确定数据流图之后，针对可重构处理器的硬件特点，在了解其各寄存器、运算器以及各功能模块的作用机制的情况下配置可重构处理器，并生成配置信息；

[0061] (3)通过微处理器将配置信息以及所需要的各种初始数据存入相应的存储器中；

[0062] (4)最后微处理器启动可重构处理器，并将配置信息及数据发送给可重构处理器；

[0063] (5)当可重构处理器完成当前任务后，发送中断信号。

[0064] 以上详细描述了本发明的优选实施方式，但是，本发明并不限于上述实施方式中的具体细节，在本发明的技术构思范围内，可以对本发明的技术方案进行多种等同变换，这些等同变换均属于本发明的保护范围。

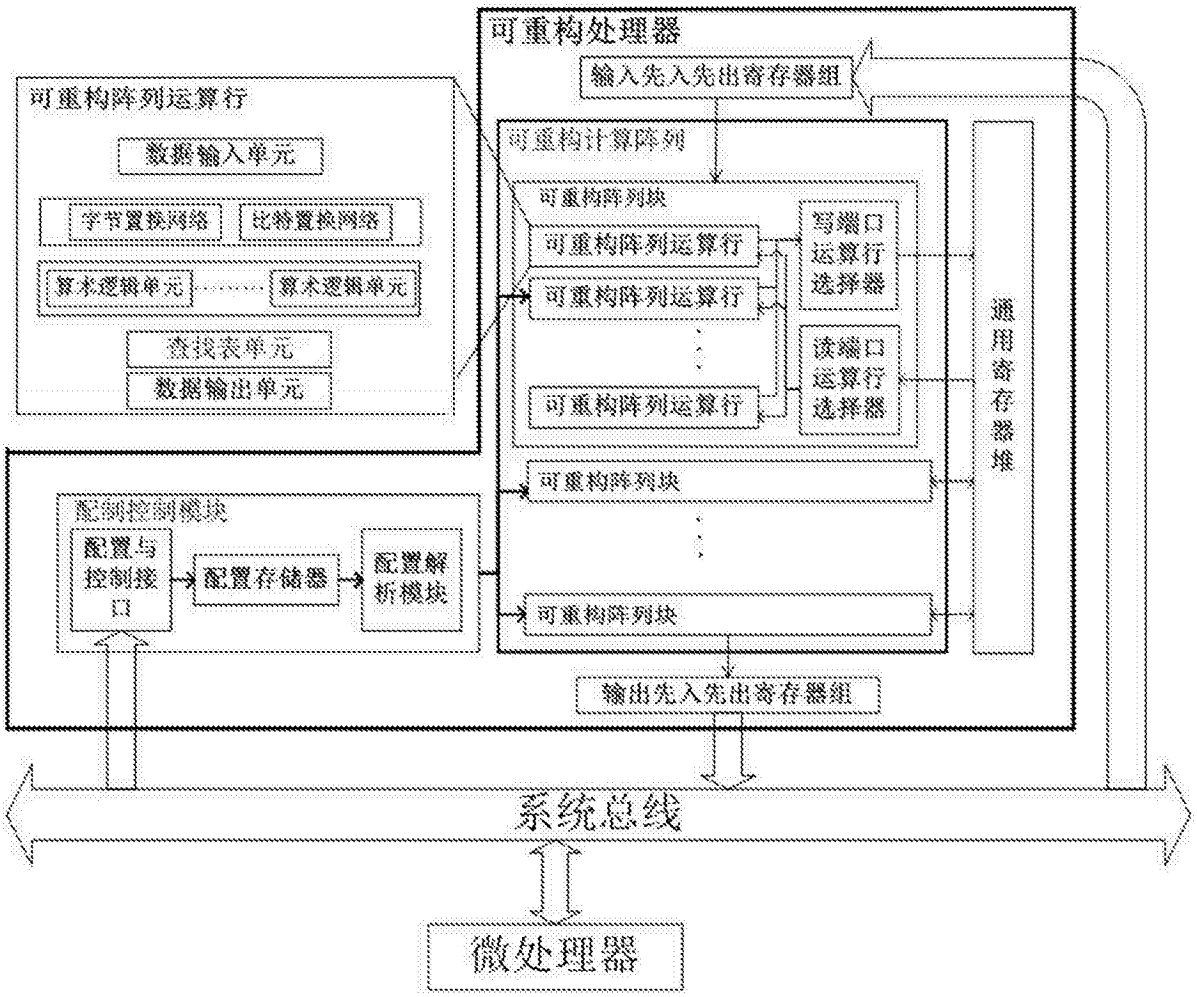

可重构处理器
输入先入先出寄存器组
可重构阵列运算行
数据输入单元
字节置换网络
比特置换网络
算术逻辑单元
算术逻辑单元
查找表单元
数据输出单元
可重构计算阵列
可重构阵列块
可重构阵列运算行
可重构阵列运算行
可重构阵列运算行
写端口运算行选择器
读端口运算行选择器
通用寄存器堆
可重构阵列块
可重构阵列块
配制控制模块
配置与控制接口
配置存储器
配置解析模块
输出先入先出寄存器组
系统总线
微处理器

图1

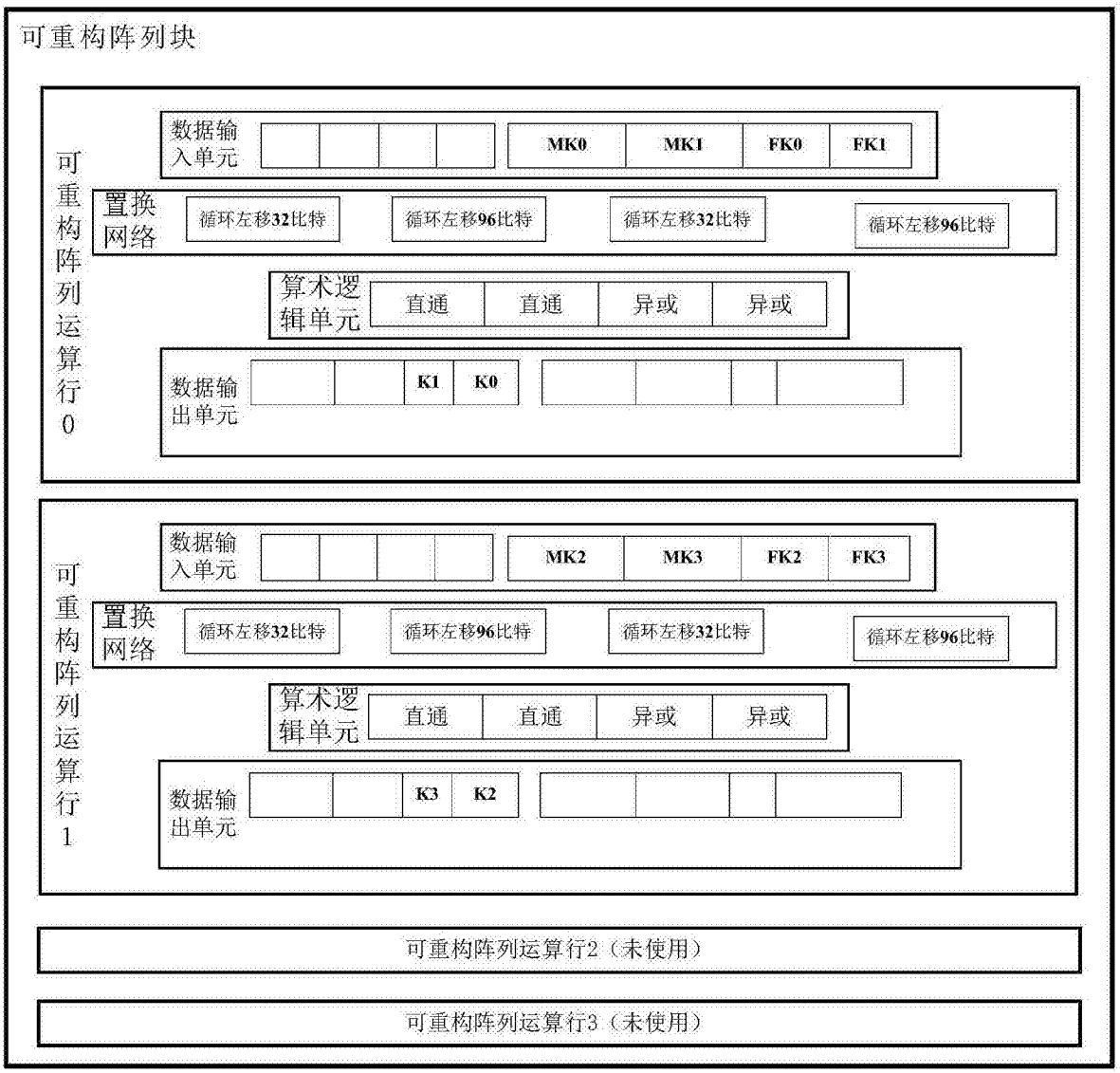

可重构阵列块
可重构阵列运算行0
数据输入单元
MK0
MK1
FK0
FK1
置换网络
循环左移32比特
循环左移96比特
循环左移32比特
循环左移96比特
算术逻辑单元
直通
直通
异或
异或
数据输出单元
K1
K0
可重构阵列运算行1
数据输入单元
MK2
MK3
FK2
FK3
置换网络
循环左移32比特
循环左移96比特
循环左移32比特
循环左移96比特
算术逻辑单元
直通
直通
异或
异或
数据输出单元
K3
K2
可重构阵列运算行2（未使用）
可重构阵列运算行3（未使用）

图2a

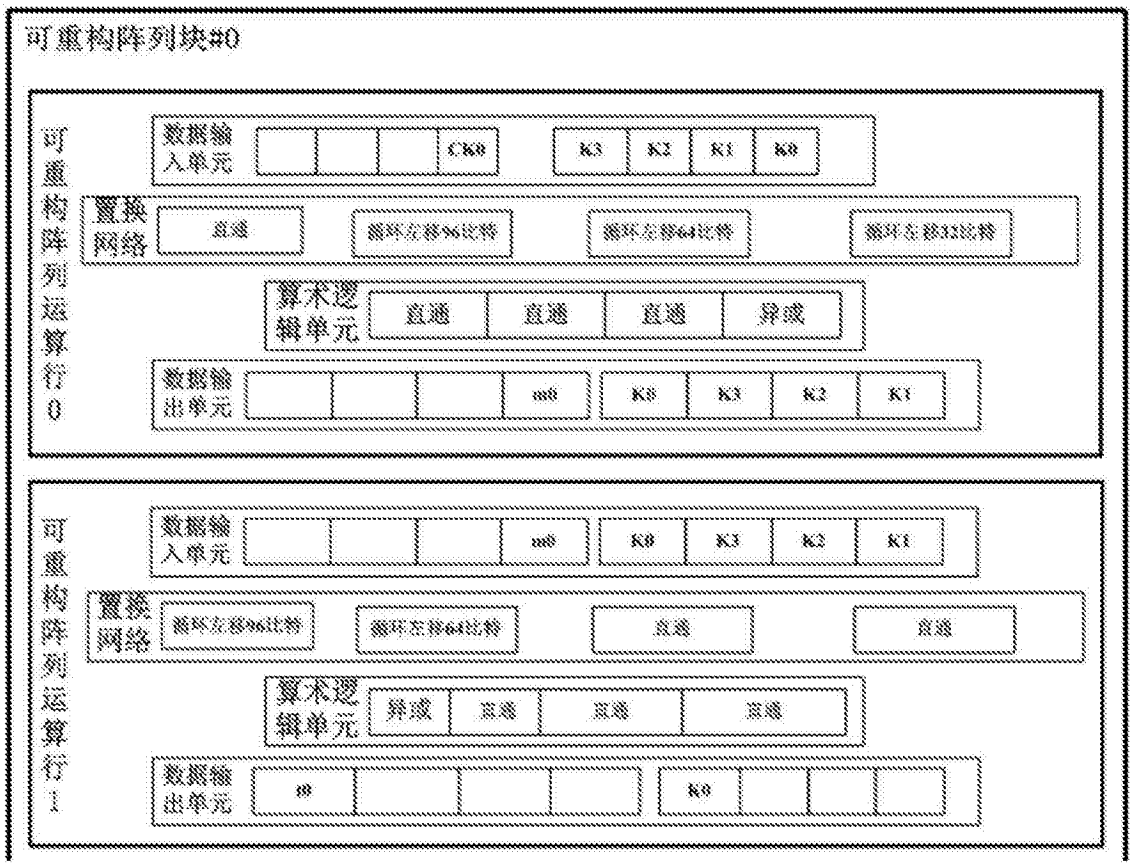

可重构阵列块#0
可重构阵列运算行0
数据输入单元
K3
K2
K1
K0
置换网络
直通
直通
直通
直通
算术逻辑单元
直通
直通
直通
异或
数据输出单元
K3
K2
K1
可重构阵列运算行1
数据输入单元
K3
K2
K1
置换网络
直通
直通
算术逻辑单元
异或
直通
直通
直通
数据输出单元
K0

图2b

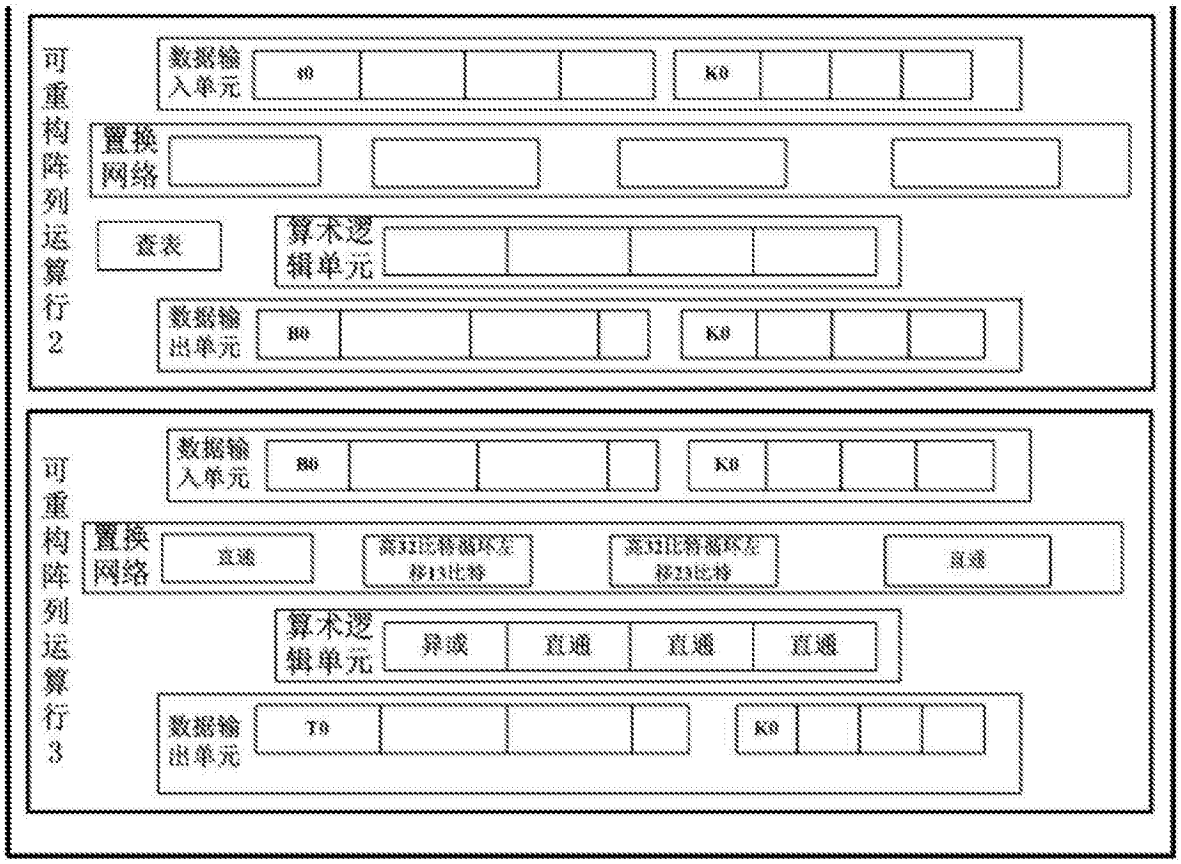

可重构阵列运算行2
数据输入单元
I0
K0
置换网络
重装
算术逻辑单元
数据输出单元
R0
K0
可重构阵列运算行3
数据输入单元
R0
K0
置换网络
直通
算术逻辑单元
异或
直通
直通
直通
数据输出单元
T0
K0

图2c

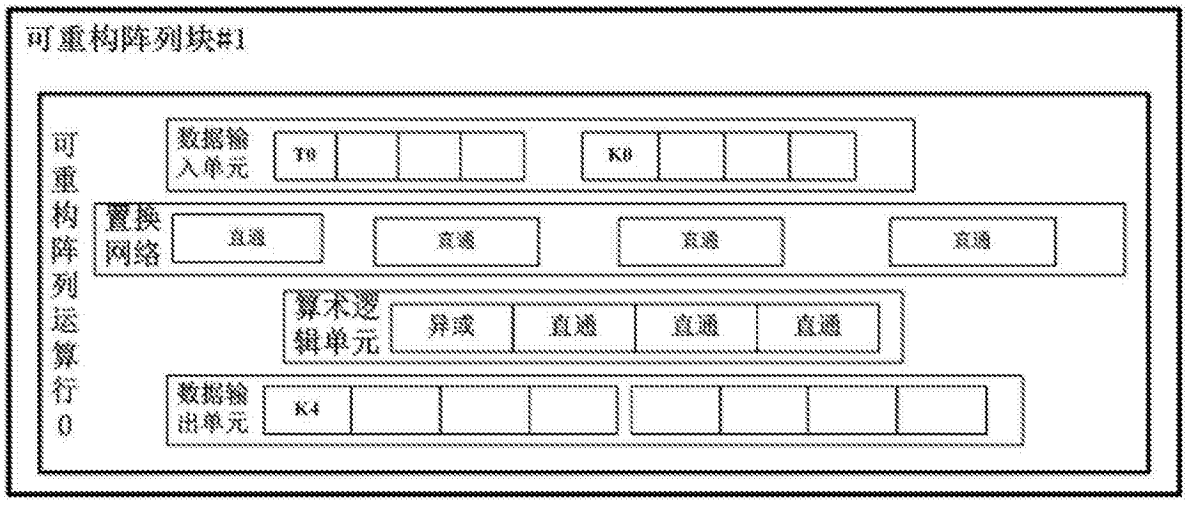

可重构阵列块#1
可重构阵列运算行0
数据输入单元
T0
K0
置换网络
直通
直通
直通
直通
算术逻辑单元
异或
直通
直通
直通
数据输出单元
K4

图2d