矿机芯片结构、算法

——国防科大2020年高性能评测与优化课程小组讨论

成员：张煊、来乐、周荣豪
指导：龚春叶、甘新标、杨博

一、研究背景

区块链

什么是比特币?

工作量证明（PoW，Proof of Work）

什么是矿机?

寻找哈希碰撞

影响挖矿收益的因素:

算力

功耗

比特币矿机发展历程

cpu

2009-2010

比特币诞生时，人人都可以使用家用电脑挖矿。挖矿门槛较低。CPU挖矿在比特币挖矿的初期大展伸手，然而在算力和难度提升的今天，传统的CPU挖矿已经渐渐被淘汰。

gpu

2010-2012

2010年，随着算力和难度不断提升，GPU挖矿正式登场。GPU的本质上是很多针对图像处理的小型集合，GPU挖矿浪费的芯片面积和功耗都比CPU少一些，算力得到了明显提升，因此挖矿效率比CPU高

fpga

2012-2013

2011年末，fpga矿机开始被用于挖矿，相比同时代的CPU、GPU矿机，FPGA虽然算力性能不占优，但功耗要低很多，综合功耗比很高。但是，随着ASIC矿机的出现，FPGA矿机逐渐淡出人们的视线。

asic

2013-now

ASIC指的是一种为专门目的而设计的集成电路，ASIC矿机就相当于专门为数字货币挖矿定制的集成电路设备，所以ASIC芯片的设计简单，成本也低，最重要的是挖矿算力也远高于同时代的CPU、GPU。ASIC矿机专门为挖矿而生，只专注于挖掘数字货币。

ASIC矿机

ASIC：专用集成电路

做SHA-256运算

成本低、速度快

比特大陆第二代7nm新品蚂蚁矿机S17Pro

S17 Pro的算力为53 TH/s，能效比为39.5 J/T。

二、比特币挖矿机制

比特币挖矿采用的是SHA256算法，针对区块头（Block Header）做二次SHA256，下图是block header的组成.

timestamp

时间戳的范围是小于当前时间+2小时，大于过去11个区块的中位数时间戳

Field	Size	Description
Version	32 bits	Block version information that is based on the Bitcoin software version creating this block
hashPrevBlock	256 bits	The hash of the previous block accepted by the Bitcoin network
hashMerkleRoot	256 bits	Bitcoin transactions are hashed indirectly through the Merkle Root
Timestamp	32 bits	The current timestamp in seconds since 1970-01-01 T00:00 UTC
bits	32 bits	The current Target represented in a 32 bit compact format
Nonce	32 bits	Goes from 0x00000000 to 0xFFFFFFFF and is incremented after a hash has been tried

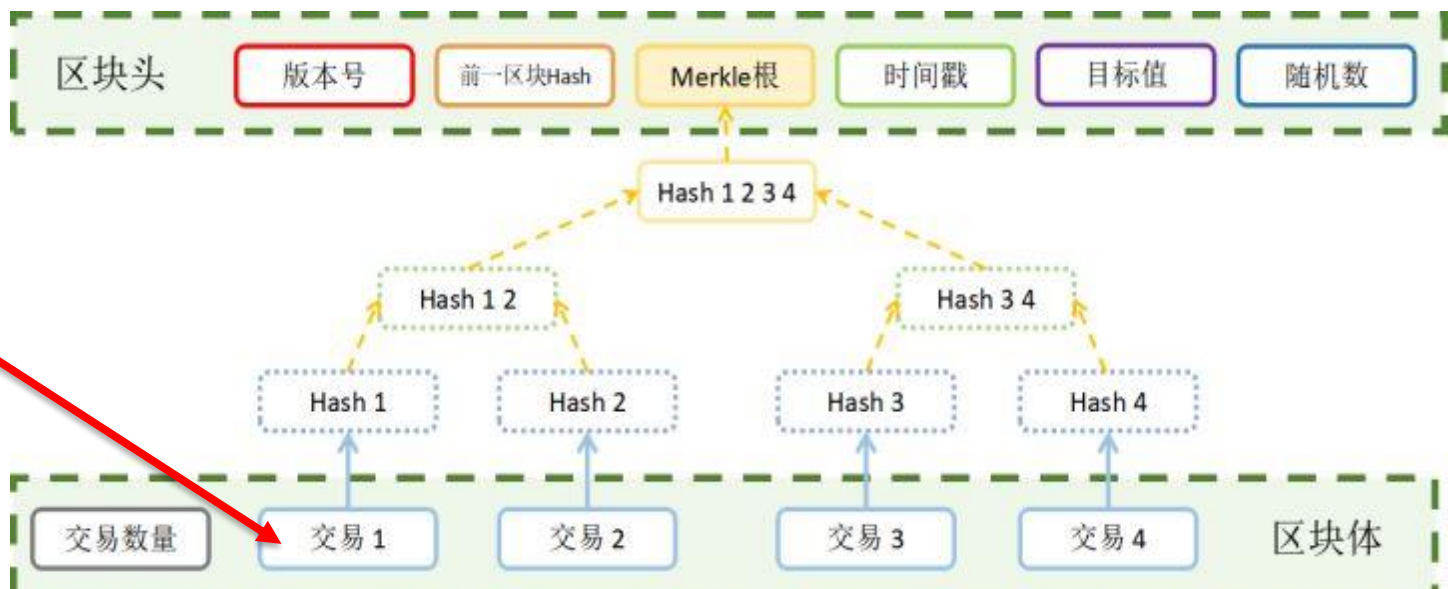
二、比特币挖矿机制

MerkleRoot

默克尔根是对这一个区块中的所有交易信息求sha256哈希.

Coinbase 交易

出块奖励和所有交易的手续费，由矿工自己建立。



二、比特币挖矿机制

矿工需要做的就是不断的修改随机数，计算出不同的hash值，当找到小于当前难度的hash值后，向全网发布，挖矿就成功了。

还需修改时间戳、默克尔根

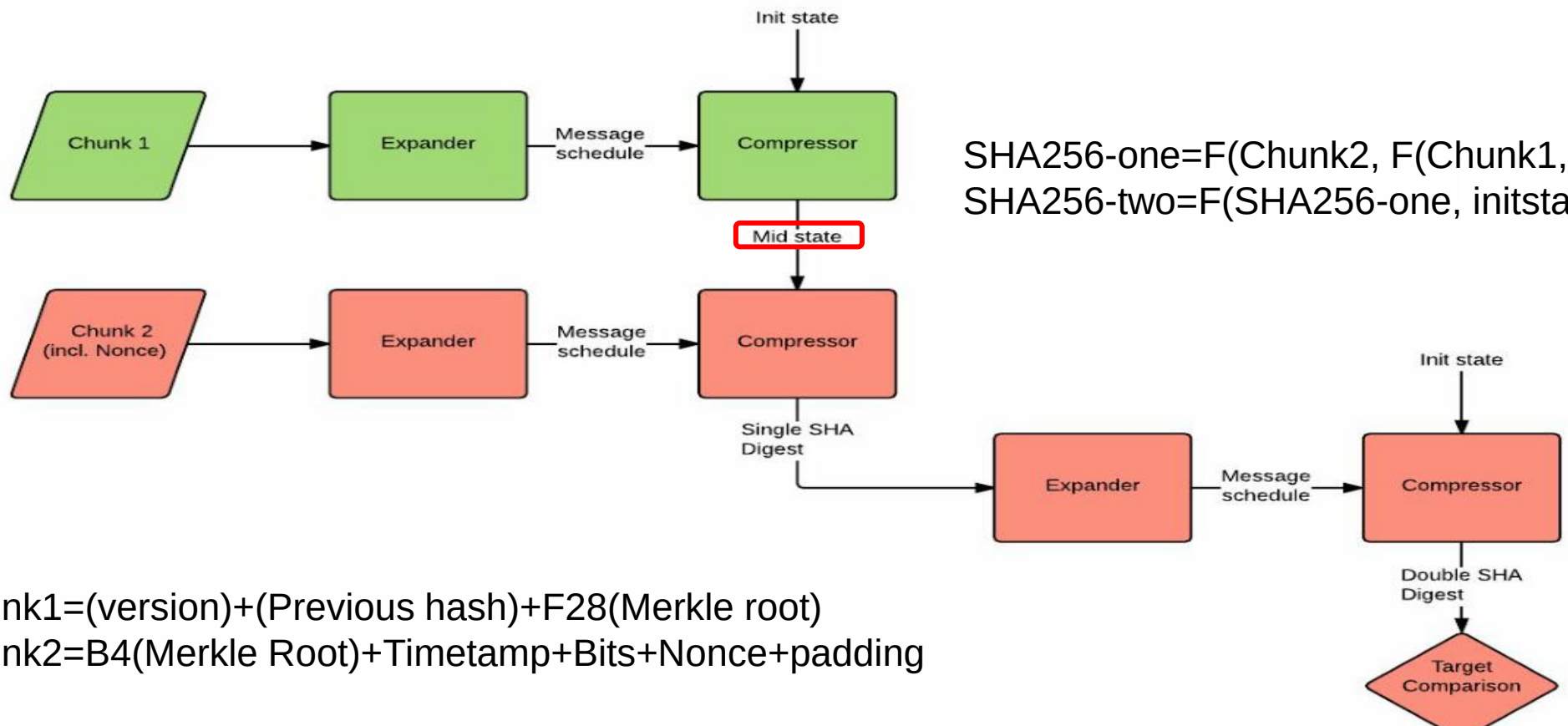
Field	Size	Description
Version	32 bits	Block version information that is based on the Bitcoin software version creating this block
hashPrevBlock	256 bits	The hash of the previous block accepted by the Bitcoin network
hashMerkleRoot	256 bits	Bitcoin transactions are hashed indirectly through the Merkle Root
Timestamp	32 bits	The current timestamp in seconds since 1970-01-01 T00:00 UTC
bits	32 bits	The current Target represented in a 32 bit compact format
Nonce	32 bits	Goes from 0x00000000 to 0xFFFFFFFF and is incremented after a hash has been tried

三、区块头的SHA-256计算

区块头做sha-256运算时的结构

<table>
<tr><td colspan="3">Chunk 1</td><td colspan="5">Chunk 2</td></tr>
<tr><td colspan="6">Block header</td><td rowspan="3">Nonce</td><td rowspan="3">Padding</td></tr>
<tr><td colspan="6">Block header candidate</td></tr>
<tr><td rowspan="2">Version</td><td rowspan="2">Previous hash</td><td colspan="2">Merkle root</td><td rowspan="2">Time stamp</td><td rowspan="2">Bits (difficulty)</td></tr>
<tr><td>Head</td><td>Tail</td></tr>
<tr><td>4 bytes</td><td>32 bytes</td><td>28 bytes</td><td>4 bytes</td><td>4 bytes</td><td>4 bytes</td><td>4 bytes</td><td>48 bytes</td></tr>
<tr><td colspan="3"></td><td colspan="3">Message[2]</td><td colspan="2"></td></tr>
</table>

三、区块头的SHA-256计算

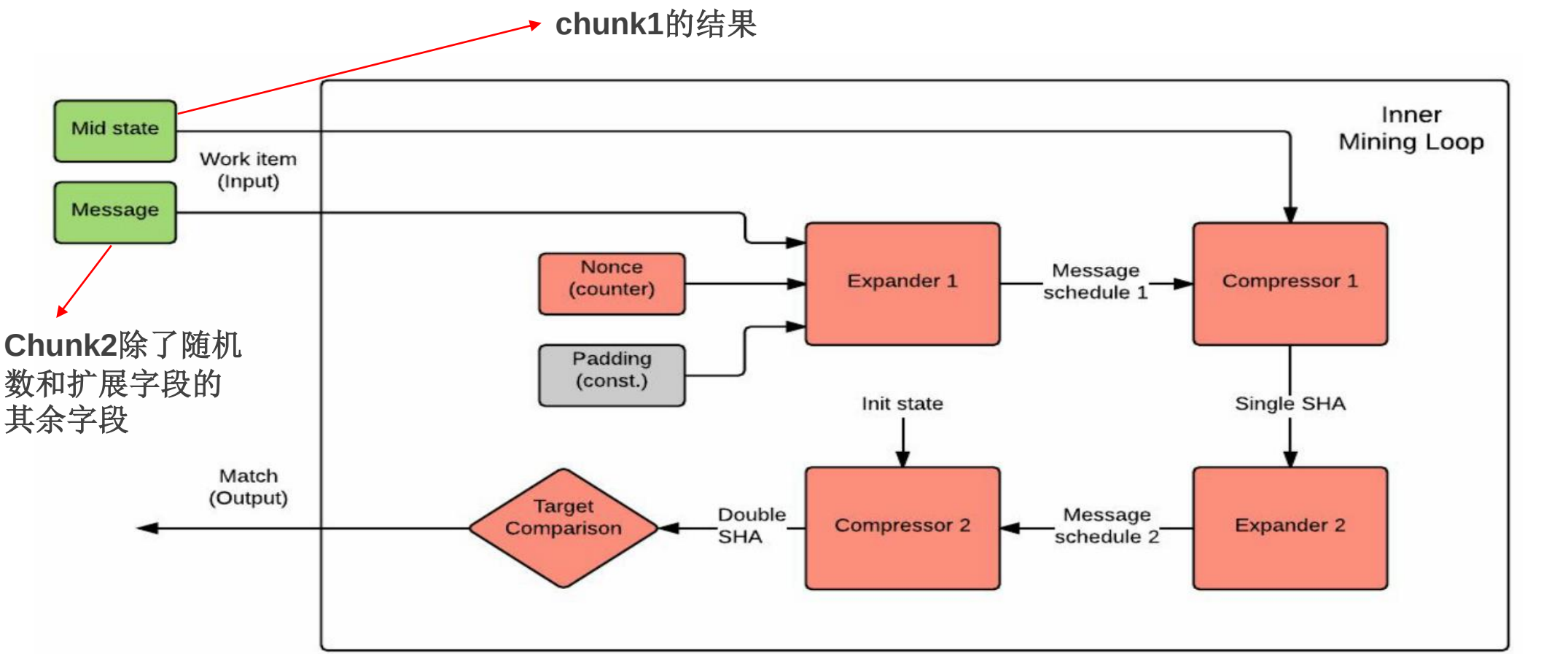


SHA256-one=F(Chunk2, F(Chunk1,initstate))
SHA256-two=F(SHA256-one, initstate)

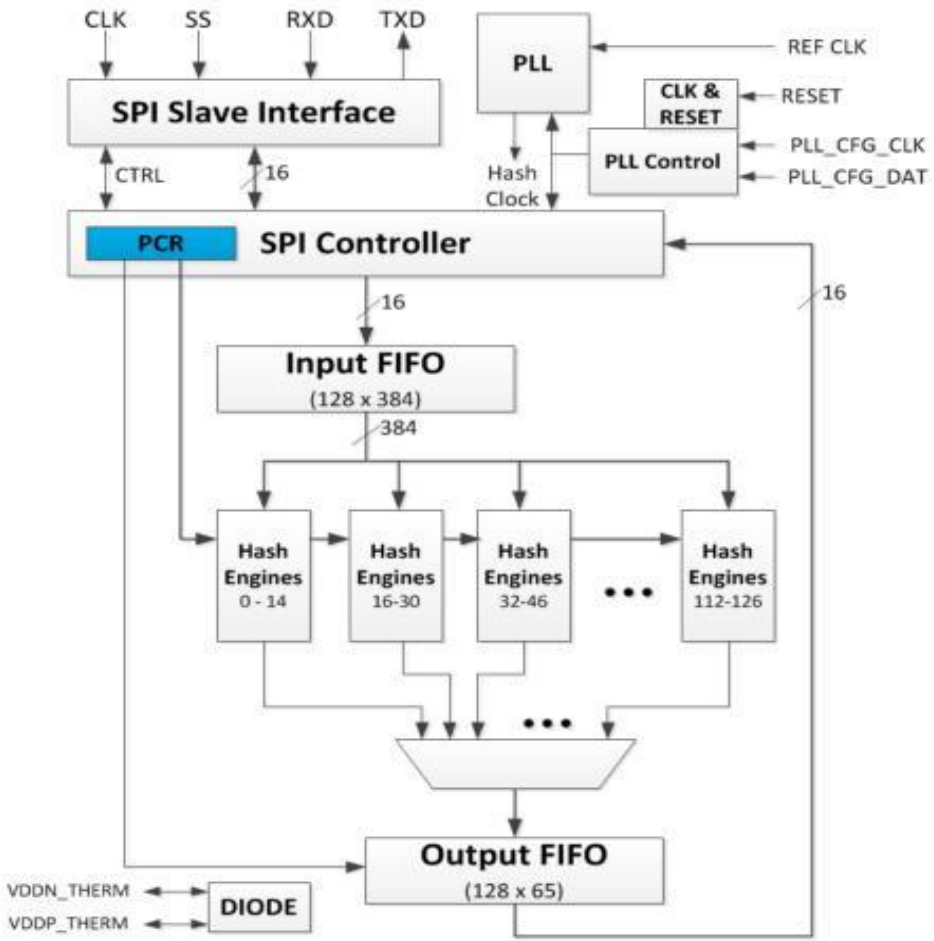
Chunk1=(version)+(Previous hash)+F28(Merkle root)
Chunk2=B4(Merkle Root)+Timetamp+Bits+Nonce+padding

三、区块头的SHA-256计算

第一区块的不变优化



四、GOLDSTRIKE芯片架构

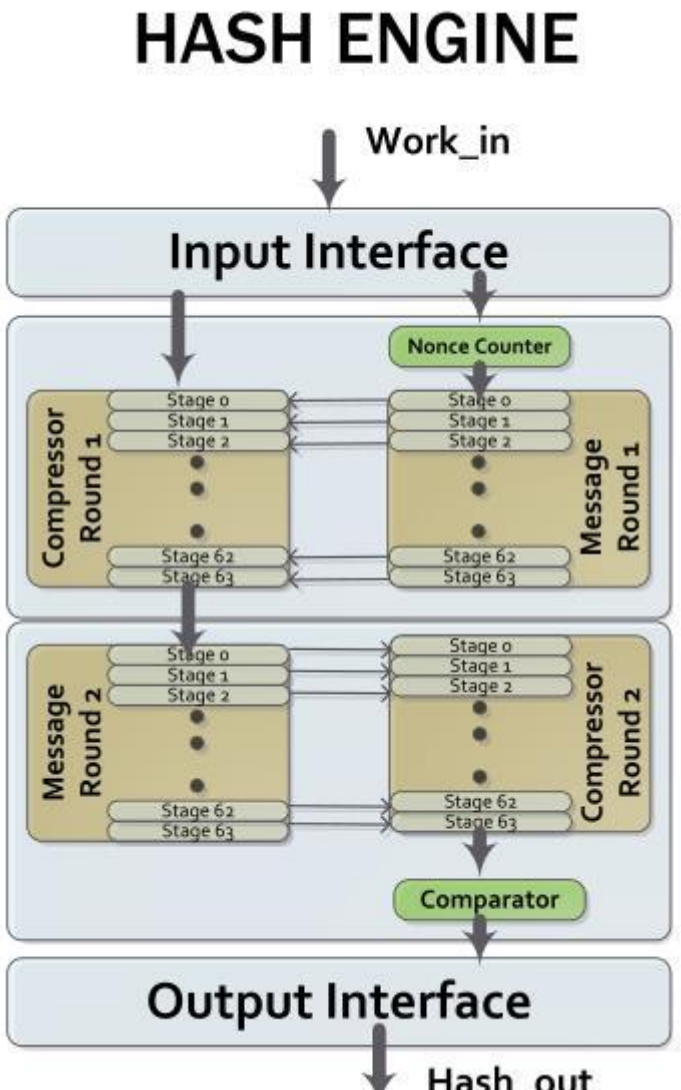


- 120个hash引擎，1.05Ghz时钟频率
  126GH/s算力
- 4针SPI接口
- 128深度,384位宽输入FIFO
  128深度,65位宽输出FIFO
- 384位pipe control register （PCR）
  用于启用/禁用任何哈希引擎
- PLL (PhaseLockedLoop)锁相环

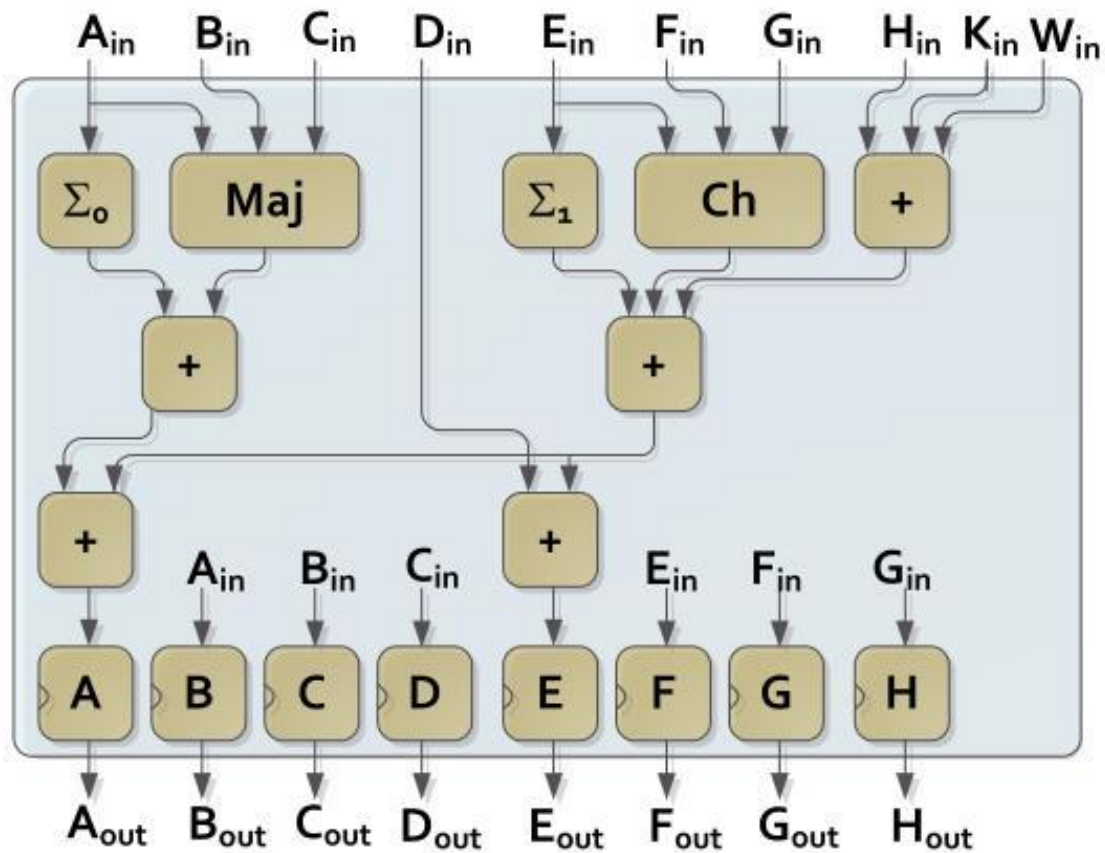
Barkatullah J , Hanke T . Goldstrike 1: CoinTerra's First-Generation Cryptocurrency Mining Processor for Bitcoin[J]. Micro, IEEE, 2015, 35(2):68-76.

四、GOLDSTRIKE芯片架构

- 经过两轮SHA-256处理
- 2^32 nonce值范围内搜索
- 每轮包含64个迭代
- 两个并行Message和 Compressor
- 仅当满足目标条件时才生成结果



COMPRESSOR STAGE OF SHA2-256



$$\Sigma_0(A) = (A \ggg s_1) \oplus (A \ggg s_2) \oplus (A \ggg s_3)$$

$$\Sigma_1(E) = (E \ggg s_4) \oplus (E \ggg s_5) \oplus (E \ggg s_6)$$

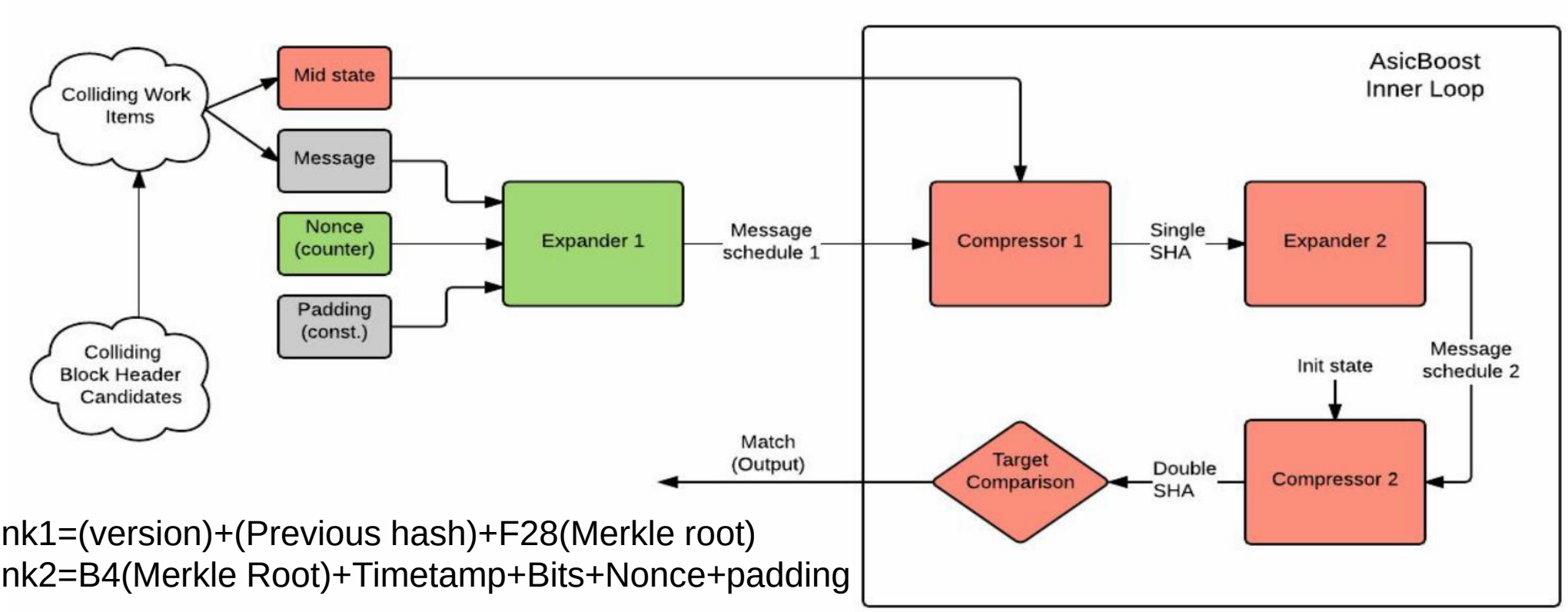
$$Ch(E,F,G) = (E \wedge F) \oplus (\sim E \wedge G)$$

$$Maj(A,B,C) = (A \wedge B) \oplus (B \wedge C) \oplus (C \wedge A)$$

All registers are 32-bits wide

五、算法优化方法 AsicBoost

白皮书：Hanke T . AsicBoost - A Speedup for Bitcoin Mining[J]. 2016.



Chunk1=(version)+(Previous hash)+F28(Merkle root)
Chunk2=B4(Merkle Root)+Timetamp+Bits+Nonce+padding

五、算法优化方法 AsicBoost

如何获得后四字节相同的默克尔根：

1）修改coinbase交易

2）交换交易的顺序

生日悖论

Bits	Possible outputs (2 s.f.) (H)	Desired probability of random collision (2 s.f.) (p)									
		10^{-18}	10^{-15}	10^{-12}	10^{-9}	10^{-6}	0.1%	1%	25%	50%	75%
16	65,536	<2	<2	<2	<2	<2	11	36	190	300	430
32	4.3×10^{9}	<2	<2	<2	3	93	2900	9300	50,000	77,000	110,000
64	1.8×10^{19}	6	190	6100	190,000	6,100,000	1.9×10^{8}	6.1×10^{8}	3.3×10^{9}	5.1×10^{9}	7.2×10^{9}
128	3.4×10^{38}	2.6×10^{10}	8.2×10^{11}	2.6×10^{13}	8.2×10^{14}	2.6×10^{16}	8.3×10^{17}	2.6×10^{18}	1.4×10^{19}	2.2×10^{19}	3.1×10^{19}
256	1.2×10^{77}	4.8×10^{29}	1.5×10^{31}	4.8×10^{32}	1.5×10^{34}	4.8×10^{35}	1.5×10^{37}	4.8×10^{37}	2.6×10^{38}	4.0×10^{38}	5.7×10^{38}
384	3.9×10^{115}	8.9×10^{48}	2.8×10^{50}	8.9×10^{51}	2.8×10^{53}	8.9×10^{54}	2.8×10^{56}	8.9×10^{56}	4.8×10^{57}	7.4×10^{57}	1.0×10^{58}
512	1.3×10^{154}	1.6×10^{68}	5.2×10^{69}	1.6×10^{71}	5.2×10^{72}	1.6×10^{74}	5.2×10^{75}	1.6×10^{76}	8.8×10^{76}	1.4×10^{77}	1.9×10^{77}

五、算法优化方法 AsicBoost

效果：

总体上能提升7%的挖矿效率

Number of colliding work items (n)	1	2	4	5	8	16
Gain in percent	0	12.5	18.75	20	21.9	23.4

SegWit（隔离见证）


THANK YOU!
感谢观看