

## طراحی پردازنده مبتنی بر FPGA برای الگوریتم‌های رمزنگاری سری SHA-2

\* ندا صدق اهرابی      \*\* محمدعلی جبرئیل جمالی

\* کارشناس ارشد، گروه مهندسی برق، دانشکده فنی مهندسی، واحد تبریز، دانشگاه آزاد اسلامی، تبریز  
\*\* دکتری، گروه مهندسی کامپیوتر، دانشکده فنی مهندسی، واحد شبستر، دانشگاه آزاد اسلامی، شبستر  
تاریخ دریافت: ۱۳۹۸/۰۴/۲۵      تاریخ پذیرش: ۱۳۹۹/۰۲/۱۳

### چکیده

الگوریتم‌های درهم‌ساز ایمن، نوعی از الگوریتم‌های رمزنگاری هستند که اهمیت آن‌ها در جامعه امروزی با بروز کاربردهایی مانند استفاده از ابزارهای دیجیتالی شخصی در راستای حفظ محرمانگی پررنگ‌تر شده‌اند. از طرفی با پیشرفت تکنولوژی، لزوم پیاده‌سازی این الگوریتم‌ها روی بسترهای انعطاف‌پذیر، می‌تواند چالش‌برانگیز باشد. کاهش مساحت و افزایش سرعت اجرای عملیات، چالش‌های اساسی برای طراحی و پیاده‌سازی این دسته از الگوریتم‌ها هستند. در این مقاله یک معماری جدید برای پردازنده مبتنی بر FPGA برای الگوریتم‌های رمزنگاری سری SHA-2 پیشنهاد شده است. در پردازنده پیشنهادی استفاده از واحدهای حافظه و مسیر داده چندپورته و به دنبال آن عملکرد موازی پردازنده باعث کاهش بکارگیری منابع و افزایش سرعت پردازش داده‌ها شده است. معماری پردازنده برای الگوریتم‌های رمزنگاری SHA-2 با زبان VHDL مدل‌سازی شده و پیاده‌سازی آن روی بستر FPGA در سری‌های Virtex توسط نرم‌افزار ISE انجام شده است. نتایج پیاده‌سازی نشان می‌دهند که پردازنده مترکم پیشنهادی در مقایسه با کارهای پیشین با اهداف مشابه، توانسته با ۲۵٪ افزایش فرکانس کاری برای الگوریتم رمزنگاری SHA-256 و اشغال ۵۵٪ مساحت کمتر برای الگوریتم رمزنگاری SHA-512 حد مطلوبی از توان عملیاتی و کارایی را نیز حفظ نماید. پردازنده پیشنهادی برای کاربردهایی مانند بسترهای سیار مورد اعتماد (TMP)، واحد پول دیجیتال (Bitcoin) و مسیریابی ایمن در شبکه روی تراشه (NoC) مناسب است.

**واژه‌های کلیدی:** الگوریتم‌های درهم‌ساز ایمن، الگوریتم‌های رمزنگاری سری SHA-2، پردازنده، VHDL، FPGA

### ۱- مقدمه

رابط‌های سیمی یا بی‌سیم در ارتباط با یکدیگرند، توزیع شده‌اند. بر همین اساس تحقیقات وسیعی مبنی بر بهبود مدیریت امنیت اطلاعات منطبق بر استانداردهای جهانی در تشکیلات گسترده، ارائه شده است [۲]. از سوی دیگر با رشد سریع تکنولوژی امنیت شبکه‌های کامپیوتری و لزوم آپلود تصاویر و اسناد شخصی در سامانه‌های مختلف، زمینه‌های تحقیقاتی وسیعی در راستای ارتقای امنیت اطلاعات در اینترنت را نیز فراهم ساخته است [۳].

رشد سریع ارتباطات الکترونیکی حاکی از آن است که مسائلی در رابطه با امنیت اطلاعات و کاربردهای عملی آن‌ها روز به روز بیشتر حائز اهمیت می‌باشند [۱]. امروزه ابزارهای محاسباتی فراگیر و ارتباطات بی‌سیم، بازگشای چالش‌های فراوانی هستند. اطلاعات حساس شخصی مانند اطلاعات پزشکی و مالی افراد در فضای وسیعی از ابزارهای دیجیتالی و محاسباتی در ارگان‌ها و سازمان‌ها، که با

امن تر یعنی خانواده SHA-2 جایگزین شده‌اند [۱۱] و [۱۲]؛ ولی با وجود پیشنهاد الگوریتم SHA-3 در سال‌های اخیر [۱۵] و [۱۳]، خانواده SHA-2 هنوز هم یک سطح امنیتی کافی را برای ابزارها فراهم می‌کند.

به موازات پیشرفت تکنولوژی و بروز کاربردهای جدید از الگوریتم‌های درهم‌ساز ایمن، حجم وسیعی از تحقیقات حول پیاده‌سازی روی بسترهای FPGA به واسطه انعطاف‌پذیری آن‌ها انجام شده است. در ابتدا قالب این پیاده‌سازی‌ها با هدف کاهش تأخیر [۱۶] و افزایش سرعت اجرای الگوریتم به واسطه کاهش مسیرهای بحرانی [۱۷]، Unfolded [۱۵] و به کارگیری روش‌های موازی [۱۸] که از بروزترین تحقیقات در این راستا می‌توان به تابع هش شبکه موازی (PLHF) اشاره کرد [۱۹]، انجام گرفته است. از طرفی سابقه پیاده‌سازی الگوریتم‌های درهم‌ساز ایمن با رویکرد کاهش سطح [۲۳] و [۲۰] با به کارگیری روش‌های Folding و ساختار موازی مرحله پیش‌پردازش الگوریتم با هدف به کارگیری در کاربردهای مهمی همچون کاربردهای مبتنی بر وب (Web-based) [۲۴] و امنیت روترهای شبکه در پیشگیری از حمله غیرقابل پیش‌بینی هکرها [۲۵] انجام شده است. آنچه معماری جدید پردازنده را از کارهای قبلی متمایز می‌سازد به کارگیری واحدهای مسیر داده و حافظه چندپورته جهت فراهم‌سازی شرایطی برای عملکرد موازی الگوریتم SHA-2 و استفاده از ثبات‌هایی برای ذخیره موقت داده‌ها جهت کاهش میزان دسترسی به حافظه در قالب یک معماری ساده است که هدف از این طرح کاهش مساحت و افزایش سرعت پردازش داده‌ها به موازات حفظ حد مطلوبی از توان عملیاتی و کارایی در کاربردهای سیار است.

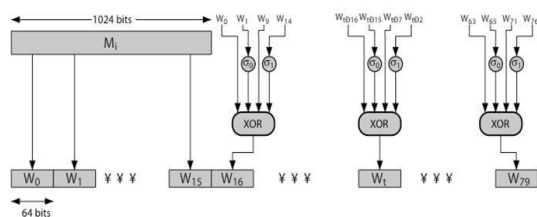
در ادامه مقاله در بخش ۲، الگوریتم SHA-2 به‌طور اجمالی تشریح شده و در بخش ۳، بلوک‌دیگرام معماری پردازنده پیشنهادی همراه با مازول‌های آن شرح داده شده است. بخش ۴، به ارزیابی طرح پیشنهادی و مقایسه نتایج حاصل از سنتز و پیاده‌سازی الگوریتم‌های رمزنگاری SHA-256 و SHA-512 پرداخته و در بخش ۵، نتیجه‌گیری کلی اعلام شده است.

## ۲- مفاهیم اولیه الگوریتم SHA-2

از دیرباز الگوریتم‌های رمزنگاری<sup>۱</sup> به‌عنوان الگوریتم‌هایی با سرعت کم، منابع محاسباتی زیاد و پیاده‌سازی‌های ناکارآمد همه‌منظوره<sup>۲</sup> مورد توجه بوده‌اند [۴] و [۵]؛ ئیزه‌ای برای پیشنهاد و پیاده‌سازی یک سرعت بالا برای الگوریتم‌های رمزنگاری ایجاد کرده است. همچنین امروزه به‌واسطه استفاده فراوان از ابزارهای سیار مانند ابزارهای دیجیتالی شخصی و گوشی‌های هوشمند، برنامه‌های کاربردی جدیدی ظهور کرده‌اند که به دنبال خود خطرهای احتمالی برای سیستم‌های مذکور خواهند داشت. در همین راستا و با توجه به این که پیشرفت تکنولوژی اهمیت پیاده‌سازی این الگوریتم‌ها روی بسترهای انعطاف‌پذیر مانند<sup>۳</sup> FPGAها [۶]، CPLDها<sup>۴</sup> و ASICها<sup>۵</sup> که سطوح امنیتی بالایی را تأمین کنند، پررنگ می‌کند [۹] و [۷]. از آنچه گفته شد می‌توان نتیجه گرفت که پیاده‌سازی سرویس‌های امنیتی که قابل مصالحه از نظر مساحت، فرکانس، توان عملیاتی ۶ و کارایی ۷ باشند، امری ضروری است. معماری پیشنهادی در قالب یک پردازنده جهت اجرای دستورالعمل‌های الگوریتم‌های رمزنگاری سری SHA-2 با رویکرد تخصیص حداقل منابع طراحی شده است؛ از طرفی پردازنده مترام مذکور مبتنی بر FPGA بوده که با استفاده از واحدهای حافظه و مسیرهاده چندپورته و استفاده از برخی امکانات موجود در بسترهای نامبرده کارهایی در راستای اجرای سریع دستورالعمل‌ها به کار گرفته شده است.

SHA مخفف عبارت Secure Hash Algorithm و به معنی الگوریتم درهم‌سازی ایمن است. این الگوریتم‌ها شامل توابعی هستند که عمل فشرده‌سازی را انجام می‌دهند؛ بدین معنی که برای ورودی‌های با طول متفاوت، کدهایی (خروجی‌هایی) با طول ثابت تولید می‌کنند که این کدها یک طرفه بوده و نمی‌توان از روی خروجی الگوریتم، ورودی آن را تشخیص داد [۱۰].

الگوریتم‌های درهم‌ساز ایمن مجموعه‌ای از توابع درهم‌سازی محسوب می‌شوند که از سال ۱۹۹۵ تاکنون در چهار نسخه توسط آژانس امنیت ملی ایالت متحده آمریکا<sup>۱</sup> (NSA) طراحی و توسط موسسه ملی فناوری و استانداردها<sup>۲</sup> (NIST) به‌عنوان استاندارد پردازش اطلاعات انتشار یافته‌اند. از سال ۲۰۱۰ نسخه‌های SHA-0 و SHA-1 با گونه‌های



شکل ۲: تابع دور الگوریتم SHA-512 [۲۶]

مقدار هش  $H_i$  از پردازش کامل بلوک اطلاعاتی  $M_i$  جاری به ترتیب ذیل به دست می‌آید.

### ۲-۱-۱- آماده‌سازی زمانبند پیام ( $W_t$ )

زمانبند پیام عبارت است از پیام‌هایی که از پیام اصلی جهت به کارگیری در عملیات مربوط به دورهای الگوریتم به دست می‌آیند؛ که در آن مقدار اول در ۱۶ دور اول مستقیماً از پیام ورودی حاصل می‌شود و برای بقیه دورها (دورهای  $17 \leq t \leq 64$  برای SHA-256 و دورهای  $17 \leq t \leq 80$  برای SHA-512) مقادیر  $W_t$  با استفاده از توابع  $\sigma_0$  و  $\sigma_1$  محاسبه می‌شوند [۱۲].

$$W_t = \begin{cases} M_t \\ \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16} \end{cases} \quad (1)$$

$$\sigma(x)_0^{256} = ROTR^7(x) \oplus ROTR^{18}(x) \oplus SHR^3(x) \quad (2)$$

$$\sigma(x)_1^{256} = ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x) \quad (3)$$

$$\sigma(x)_0^{512} = ROTR^1(x) \oplus ROTR^8(x) \oplus SHR^7(x) \quad (4)$$

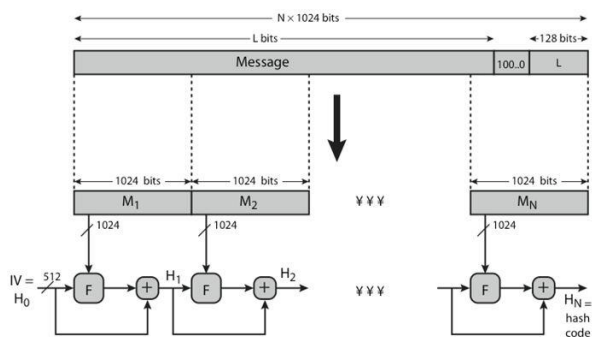
$$\sigma(x)_1^{512} = ROTR^{19}(x) \oplus ROTR^{61}(x) \oplus SHR^6(x) \quad (5)$$

که در روابط فوق منظور از  $ROTR^i(x)$  چرخش راست  $i$  واحدی عبارت  $x$  و منظور از عبارت  $SHR^i(x)$  انتقال به راست  $i$  واحدی عبارت  $x$  است.

### ۲-۱-۲- مرحله فشرده‌سازی

در این مرحله حلقه اصلی پردازش بلوک اطلاعاتی  $M_i$  از ۸ متغیر  $a, b, c, d, e, f, g, h$  در قالب بافرهای حالت بوده و پس از طی شدن دورهای داخلی الگوریتم به‌عنوان مقدار هش میانی  $Htemp$  در نظر گرفته می‌شود. این مقدار هش با مقدار محاسبه شده برای بلوک اطلاعاتی قبلی جمع شده و

سری دوم الگوریتم‌های درهم‌ساز ایمن، مجموعه‌ای از توابع درهم‌ساز هستند که بر اساس طول چکیده (به تعداد بیت) در قالب‌های کلی به دو صورت SHA-256 و SHA-512 نام‌گذاری شده‌اند که محاسبات آن‌ها به ترتیب با کلمات ۳۲ و ۶۴ بیتی صورت می‌گیرد. این دو الگوریتم، ساختار نزدیک به هم دارند ولی در تعداد دورها و مقادیر اولیه به‌کاررفته متفاوت هستند که از بین آن‌ها استفاده از SHA-512 از نظر میزان امنیت حاصله و سرعت آن روی سیستم‌های ۶۴ بیتی ارجحیت دارد. در شکل ۱ ساختار کلی الگوریتم SHA-512 نشان داده شده است که در ادامه عملیات اجرایی مربوط به شکل توضیح داده شده است.

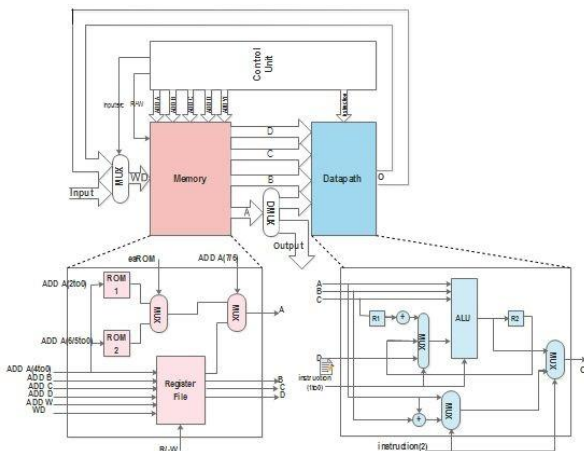


شکل ۱: عملیات مربوط به الگوریتم رمزنگاری SHA-512 [۲۶]

### ۲-۱-۲- مرحله پیش‌پردازش

طول پیام ورودی (Message) با افزودن عدد '۱' و به دنبال آن صفرهایی برای رسیدن به مضرری از ۵۱۲ در SHA-256 (۱۰۲۴ در SHA-512) افزایش می‌یابد. ۶۴ بیت آخر پیام در SHA-256 (۱۲۸ بیت آخر پیام در SHA-512) برای ذخیره طول پیام اصلی استفاده می‌شود. بعد از عمل افزودگی، پیام به‌دست آمده به بلوک‌هایی به طول ۵۱۲ بیتی برای SHA-256 (۱۰۲۴ بیتی برای SHA-512) به صورت  $M_1, M_2, \dots, M_N$  تقسیم می‌شود. هر بلوک از اطلاعات  $M_i$  به ترتیب با توابع اصلی ( $F$ ) در طول ۶۴ دور برای SHA-256 و ۸۰ دور برای SHA-512 پردازش می‌شوند. تابع دور مربوط به الگوریتم SHA-512 در شکل ۲ نشان داده شده است.

معماری پردازنده پیشنهادی از سه واحد مسیر داده، حافظه و کنترل تشکیل شده است. ایده اصلی معماری پیشنهادی استفاده از مسیر داده و حافظه چندپورته جهت عملکرد موازی چند مرحله از الگوریتم SHA-2، تغییر در روابط الگوریتم با حفظ ماهیت آن‌ها به منظور کاهش مسیره‌های بحرانی، استفاده از ثبات‌های موقت در واحد مسیر داده برای کم کردن میزان دسترسی به واحد حافظه و استفاده از بانک ثبات به منظور استفاده مجدد از منابع حافظه و تسریع دسترسی به اطلاعات است. همچنین امکان نگاشت واحد محاسباتی و منطقی روی جداول جستجوی ۱۰ چهار ورودی در بسترهای FPGA، امکان پیاده‌سازی طرح پیشنهادی روی این بسترها را تسهیل می‌کند. بلوک‌دیگرام پردازنده هش در شکل ۳ نشان داده شده است.



شکل ۳: بلوک دیگرام پردازنده هش پیشنهادی

### ۳-۱- مازول‌های بکاررفته در پردازنده هش

#### پیشنهادی

- مسیر داده

واحد مسیر داده همان‌طور که در شکل ۱ نشان داده شده است از تعدادی ثبات به منظور ذخیره موقتی داده‌ها و از یک واحد محاسباتی و منطقی برای اجرای عملیات مربوط به دوره‌های داخلی الگوریتم SHA-2 تشکیل شده است. در این پردازنده عملیات مربوط به دوره‌های داخلی الگوریتم-SHA-2 به صورت روابط  $T'$  و  $T''$  (روابط ۱۶-۱۴) جهت کاهش مسیره‌های بحرانی بازنویسی شده‌اند.

$$T = \sum_1(e) + ch(e, f, g) + W_t \quad (15)$$

مقدار هش مربوط به پیام تا بلوک اطلاعات  $M_i$  را نتیجه می‌دهد. وقتی بلوک اطلاعاتی اولیه پردازش می‌شود، مقدار هش اولیه  $H_0$  مورد استفاده قرار می‌گیرد. در اولین مرحله از محاسبه  $H_{i+1}$  بایستی مقدار  $H_i$  وارد بافرهای حالت شوند. عملیات مربوط به این مرحله در رابطه (۶) نشان داده شده‌اند.

$$\begin{aligned} h &= g & , g &= f \\ f &= e & , e &= d + T_1 \\ d &= c & , c &= b \\ b &= a & , a &= T_1 + T_2 \end{aligned} \quad (6)$$

$$H_{temp} = a | b | c | d | e | f | g | h$$

$$H^{(i)} = H^{(i-1)} + H_{temp}$$

متغیرهای زمانی به کاررفته در الگوریتم در قالب رابطه‌های (۷) و (۸) محاسبه می‌شوند [۱۲].

$$T_1 = h + \sum_1(e) + ch(e, f, g) + k_t + W_t \quad (7)$$

$$T_2 = \sum_0(a) + Maj(a, b, c) \quad (8)$$

مقادیر هش اولیه  $H_0$  و ثابت  $K_t$  به کاررفته در روابط فوق، در مشخصات و ویژگی‌های توابع SHA-256 و SHA-512 قابل دسترسی است.

تمامی عملیات درگیر در الگوریتم بر اساس روابط (۹) الی (۱۴) محاسبه می‌شوند [۲۶] و [۲۷].

$$ch(x, y, z) = (x \wedge y) \oplus (\neg x \wedge z) \quad (9)$$

$$Maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \quad (10)$$

$$\sum_0^{256}(x) = ROTR^2(x) \oplus ROTR^{13}(x) \oplus ROTR^{22}(x) \quad (11)$$

$$\sum_1^{256}(x) = ROTR^6(x) \oplus ROTR^{11}(x) \oplus ROTR^{25}(x) \quad (12)$$

$$\sum_0^{512}(x) = ROTR^{28}(x) \oplus ROTR^{34}(x) \oplus ROTR^{39}(x) \quad (13)$$

$$\sum_1^{512}(x) = ROTR^{14}(x) \oplus ROTR^{18}(x) \oplus ROTR^{41}(x) \quad (14)$$

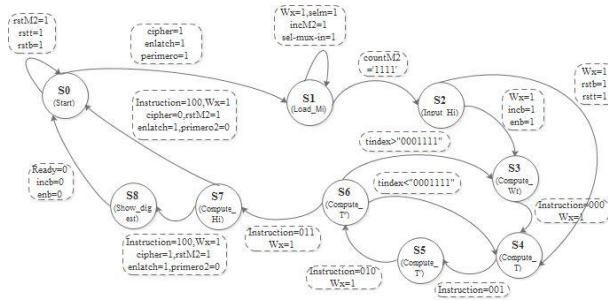
بعد از محاسبه بلوک اطلاعاتی نهایی  $M_N$  مقدار هش نهایی  $H_N$  محاسبه می‌شود.

### ۳- پردازنده متراکم پیشنهادی

متغیرهای زنجیره‌ای اولیه و ۶۴ خانه‌ای برای SHA-256 (۱۲۸ خانه‌ای برای SHA-512) به منظور ذخیره ثابت‌های دور  $K_f$  در واحد حافظه به کار گرفته می‌شوند.

- واحد کنترل

واحد کنترل از چند واحد شمارنده به منظور آدرس‌دهی بانک ثابت و ROMها تشکیل شده است؛ همچنین واحد کنترل با تولید تعدادی سیگنال کنترلی منابع اطلاعاتی واحد محاسباتی و منطقی را نیز هماهنگ می‌کند. در طول ۱۶ سیکل ساعت اول اطلاعات ورودی که شامل ۱۶ کلمه ۳۲ بیتی برای SHA-256 و ۱۶ کلمه ۶۴ بیتی برای SHA-512 هستند، جهت استفاده در توابع فشرده‌سازی به کار گرفته می‌شوند. پس از ورود کامل اطلاعات ۵۱۲ بیتی برای SHA-256 و ۱۰۲۴ بیتی برای SHA-512 واحد کنترل مقادیر میانی  $H_i$  را به بافرهای حالت انتقال می‌دهد (ثبات‌های a-h). برای اولین بلوک مقادیر هش اولیه  $H_0$  که در حافظه ثابت‌ها ذخیره شده به عنوان مقادیر هش میانی مورداستفاده قرار می‌گیرند و برای دورها و بلوک‌های اطلاعاتی بعدی،  $H_i$  توسط ثابت‌های a تا h و مقادیر هش به‌دست‌آمده از مرحله قبلی  $H_{i-1}$  محاسبه می‌شود. واحد کنترل در پردازنده هش پیشنهادی یک FSM بوده که شامل ۹ حالت است. در شکل ۵ ماشین حالت محدود واحد کنترل نشان داده شده است.



شکل ۵: ماشین حالت محدود واحد کنترل پردازنده هش پیشنهادی

#### ۴.۴ ارزیابی نتایج

در این مقاله به منظور ارزیابی نتایج حاصل از پیاده‌سازی الگوریتم‌های هش SHA-256 و SHA-512 از یک معماری مشترک (شکل ۱) برای هر کدام از توابع استفاده شده است. پس از نوشتن معماری مذکور با زبان

(

$$T' = h + k_i + d + T \quad (16)$$

$$T'' = \sum_0(a) + Maj(a,b,c) + (T' - d) \quad (17)$$

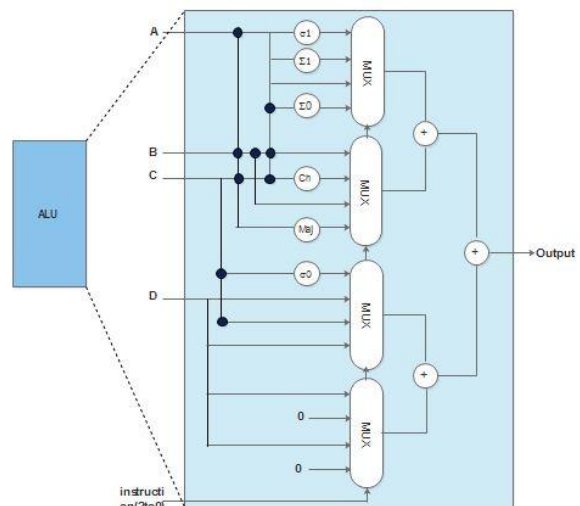
(

با توجه به ساختار الگوریتم و روابط بازنویسی شده می‌توان رابطه (۱۸) را نتیجه گرفت.

$$e = T' \quad , a = T'' \quad (18)$$

(

در شکل ۴ بلوک دیاگرام واحد محاسباتی و منطقی نشان داده شده است. ملاحظه می‌شود که به واسطه به‌کارگیری ثابت‌های موقت و بازنویسی روابط، میزان دسترسی به حافظه کم شده و از آدرس‌دهی غیرضروری به حافظه و عملیات منطقی مربوطه اجتناب شده است.



شکل ۴: بلوک دیاگرام واحد محاسباتی و منطقی

- واحد حافظه

واحد حافظه پردازنده پیشنهادی الگوریتم SHA-2 از یک بانک ثابت و دو حافظه فقط خواندنی (ROM) تشکیل شده است. بانک ثابت شامل ۳۲ ثابت ۳۲ بیتی برای SHA-256 و ۳۲ ثابت ۶۴ بیتی برای SHA-512 است که برای خواندن اطلاعات از پیش ذخیره شده و نوشتن اطلاعات حاصل از عملیات به کار می‌رود. این اطلاعات شامل متغیرهای زنجیره‌ای، کلمات اولیه و پیام‌های زمان‌بندی محاسبه شده در دورهای مختلف توسط واحد مسیر داده هستند. حافظه فقط خواندنی ۸ خانه‌ای برای نگه‌داری

۱/۱۷	SHA-512 Virtex-5
۳/۳۵	SHA-512 Virtex-7

علی‌رغم دشوار بودن مقایسه پیاده‌سازی‌های مختلف روی FPGA ها به دلیل تکنولوژی‌های مختلف به‌کاررفته، در ادامه سعی بر مقایسه عادلانه نتایج حاصله از معماری پیشنهادی با کارهای پیشین در شرایط یکسان شده است.

#### ۴-۱- نتایج به‌دست‌آمده و مقایسه‌های مربوط به الگوریتم SHA-256

تأخیر پردازنده SHA-256 برای یک بلوک داده ۵۱۲ بیتی ۲۸۰ سیکل ساعت است که به ترتیب ذیل در نظر گرفته می‌شود [۲۸]:

زمان لازم برای وارد شدن یک بلوک اطلاعاتی ۵۱۲ بیتی، ۱۶ سیکل زمانی است.

زمان لازم برای انتقال مقادیر هش میانی و بافرها (ثبات‌های a تا h) ۸ سیکل زمانی است.

زمان لازم جهت طی شدن ۶۴ دور داخلی با در نظر گرفتن دو حالت زیر محاسبه می‌شود:

برای ۱۶ دور اول،  $Wt$  مستقیماً از پیام ورودی نتیجه می‌شود که این مرحله فقط نیاز به ۳ سیکل زمانی دارد.

برای ۴۸ دور بعدی،  $Wt$  با استفاده از واحد محاسباتی و منطقی محاسبه شده و به ازای هر ۴ سیکل زمانی یک دور نتیجه می‌دهد.

بعد از محاسبات مربوط به دورهای داخلی، محاسبه مقادیر هش میانی نیاز به ۸ سیکل زمانی دارد.

نهایتاً زمان موردنیاز برای به‌دست آوردن خروجی (مقدار هش نهایی) ۸ سیکل زمانی خواهد بود.

در جدول ۲ نتایج به‌دست‌آمده از معماری پیشنهادی برای SHA-256 با کارهای پیشین مقایسه شده است. مشاهده می‌شود که در مقایسه با معماری متراکم در [۳۰] علاوه بر

کاهش ۳۰٪ استفاده از منابع، میزان تأخیر نیز به میزان قابل توجهی کاهش یافته است. نتایج حاصل از طرح متراکم

در [۳۱] به دلیل پیاده‌سازی روی بستر دیگری از خانواده Virtex با معماری پیشنهادی قابل مقایسه نیست، ولی در

حالت کلی امتیازهای این طرح از نتایج به‌دست‌آمده مشخص است. طرح پیشنهادی در مقایسه با معماری‌های

توصیف سخت‌افزاری VHDL نتایج مربوط به متغیرها در هر دور با نرم‌افزار ISIM قابل دسترس بوده و سنتز و پیاده‌سازی آن برای تعدادی از FPGA های شرکت Xilinx توسط نرم‌افزار Xilinx ISE مورد بحث و بررسی قرار می‌گیرد.

FPGA های انتخابی جهت بررسی نتایج ذکرشده به ترتیب ذیل است:

۱- Virtex-4 (Xc4vlx100-12FF1148)

۲- Virtex-5 (Xc5vlx155t-3FF1136)

۳- Virtex-7 (Xc7vx330t-3FFG1157)

طرح VHDL معماری پردازنده پیشنهادی در هر مرحله (در هر دور از الگوریتم) از نظر جریان داده و بررسی حالات

هر متغیر به‌واسطه مقادیر به‌دست‌آمده از نتایج شبیه‌سازی توسط نرم‌افزار ISIM با نتایج ذکرشده در FIPS 180\_2

[۱۱] مقایسه و اشکال‌زدایی شده و نهایتاً با این کار برای محاسبه مقادیر هش یک عبارت معتبرسازی می‌شود. در این

مقاله کیفیت معماری پیشنهادی با معیارهای مساحت، فرکانس، توان عملیاتی و کارایی موردتوجه قرار گرفته است

که پارامترهای مساحت و فرکانس مستقیماً از گزارش‌های حاصله از سنتز و پیاده‌سازی و توان عملیاتی و کارایی

به‌صورت روابط (۱۹) و (۲۰) محاسبه می‌شوند [۲۹] و [۲۸].

(۱۹) (زمان تأخیر/فرکانس) × اندازه بلوک داده = توان عملیاتی

(۲۰) مساحت/توان عملیاتی = کارایی

( )

نتایج به‌دست‌آمده از سنتز و پیاده‌سازی الگوریتم‌های SHA-256 و SHA-512 برای پارامتر کارایی در جدول

۱ و برای پارامترهای فرکانس، مساحت و توان عملیاتی در جدول‌های ۲ و ۳ در مقایسه با کارهای پیشین نشان

داده شده است.

#### جدول ۱: کارایی حاصله از پردازنده پیشنهادی

عنوان چیپ	کارایی (Mbps/Slice)
SHA-256 Virtex-4	۰/۲۷
SHA-256 Virtex-5	۱/۴۱
SHA-256 Virtex-7	۶/۹۸
SHA-512 Virtex-4	۰/۲۰



برای ۶۴ دور بعدی، Wt با استفاده از واحد محاسباتی و منطقی محاسبه شده و به ازای هر ۴ سیکل زمانی یک دور نتیجه می‌دهد.

بعد از محاسبات مربوط به دورهای داخلی، محاسبه مقادیر هش میانی نیاز به ۸ سیکل زمانی دارد. نهایتاً زمان مورد نیاز برای به دست آوردن خروجی (مقدار هش نهایی) ۸ سیکل زمانی است.

در جدول ۳ نتایج به دست آمده از معماری پیشنهادی برای SHA-512 با کارهای پیشین مقایسه شده است. معماری متراکم پیشنهادی می‌تواند جایگزین مناسبی برای [۳۴] که با هدف افزایش توان عملیاتی طراحی و پیاده‌سازی شده، روی بستر Virtex-5 باشد. از نتایج به دست آمده در [۳۵] که در آن از روش موازی با رویکرد افزایش فرکانس استفاده شده، می‌شود که در طرح متراکم پیشنهادی علاوه بر بهبود ۲۰٪ فرکانس، از ۶۰٪ منابع کمتری روی Virtex-5 استفاده شده است. هسته هشت تجاری در [۳۶] برای اجرای الگوریتم SHA-512 از ۱/۷ برابر منابع کمتری نسبت به طرح پیشنهادی استفاده می‌کند ولی برای کاربردهایی که در آن‌ها نیاز به توان عملیاتی بالا وجود دارد، پیاده‌سازی طرح پیشنهادی روی بستر Virtex-5 به دلیل بهبود ۱۱ برابری توان عملیاتی ارجحیت دارد. مشاهده می‌شود که در معماری پیشنهادی کاهش ۶۸٪ پارامتر مساحت در سطح فرکانس مطلوب، نسبت به معماری [۳۷] روی بستر Virtex-4 حاصل شده است؛ همچنین در [۳۸] به کارگیری از روش‌های unrolling و pipeline، استفاده مجدد از منابع و پیش‌محاسبات زمانی با رویکرد افزایش فرکانس و توان عملیاتی انجام شده است؛ مشاهده می‌شود که معماری پیشنهادی پیاده‌سازی شده روی بستر Virtex-5 توانسته از ۵۵٪ منابع کمتری در همان سطح فرکانسی استفاده کند.

جدول ۳: نتایج مقایسه حاصله از سنتز و پیاده‌سازی

پردازنده SHA-512

منابع	بستر	مساحت (Slices)	فرکانس (MHz)	تأخیر	توان عملیاتی (MBPS)
[۳۴]	Virtex-2	۱۹۳۸	۸۱	—	۲۷۴
[۳۵]	Virtex-4	۲۰۷۳	۱۰۶/۶۵	—	—
	Virtex-5	۱۱۰۲	۱۴۲/۸۸	—	—
[۳۶]	Hash Core	۲۵۱	۲۷۱	—	۴۶

[۳۲،۳۳] که با هدف کاهش تأخیر و افزایش توان عملیاتی طراحی شده‌اند، توانسته به موازات کاهش قابل توجه مساحت به افزایش ۱۰٪ فرکانس نیز دست یابد. همچنین در مقایسه با [۲۸] علی‌رغم اشغال ۷٪ مساحت بیشتر در معماری پیشنهادی، ۲۵٪ افزایش فرکانس و به دنبال آن بهبود توان عملیاتی و کارایی در نتایج به دست آمده از پیاده‌سازی معماری پیشنهادی روی بستر Virtex-4 حاصل شده است.

جدول ۲: نتایج مقایسه حاصله از سنتز و پیاده‌سازی

پردازنده SHA-256 پیشنهادی

منابع	بستر	مساحت (Slices)	فرکانس (MHz)	تأخیر	توان عملیاتی (MBPS)
[۳۰]	Virtex-4	۶۱۵	۱۰۲	۱۱۲۰	—
[۳۱]	Virtex-2	۱۲۱۰	۸۵	۳۵۵	۱۲۲/۶۰
[۳۲]	Virtex-5	۲۷۹۶	۱۷۹/۸	—	—
[۳۳]	Virtex-5	۱۸۸۵	۱۶۹	—	—
[۲۸]	Virtex-4	۴۲۲	۵۰/۰۶	۲۸۰	۹۱/۵۳
	Virtex-5	۱۳۹	۶۴/۴۵	۲۸۰	۱۱۷/۸
طرح پیشنهادی	Virtex-4	۴۵۳	۶۷/۱۴۶	۲۸۰	۱۲۲/۷۸
	Virtex-5	۲۵۲	۱۹۴/۹۲	۲۸۰	۳۵۶/۴۲
	Virtex-7	۷۷	۲۹۳/۷۲	۲۸۰	۵۳۷/۰۹

۲-۴ نتایج به دست آمده و مقایسه‌های مربوط به

### الگوریتم SHA-512

تأخیر پردازنده SHA-512 برای یک بلوک داده ۱۰۲۴ بیتی ۳۴۴ سیکل ساعت است که به ترتیب ذیل در نظر گرفته می‌شود:

زمان لازم برای وارد شدن یک بلوک اطلاعاتی ۱۰۲۴ بیتی، ۱۶ سیکل زمانی است.

زمان لازم برای انتقال مقادیر هش میانی و بافرها (ثبات‌های a تا h) ۸ سیکل زمانی است.

زمان لازم جهت طی شدن ۸۰ دور داخلی با در نظر گرفتن دو حالت زیر محاسبه می‌شود:

برای ۱۶ دور اول، Wt مستقیماً از پیام ورودی نتیجه می‌شود که این مرحله فقط نیاز به ۳ سیکل زمانی دارد.

محرمانگی اطلاعات در تکنولوژی روز با کاربردهایی که در آن‌ها نرخ انتقال اطلاعات زیاد است، استفاده کرد.

#### منابع

۱. ذاکر حسینی، ملکیان، امنیت داده‌ها، ویراسته ۱. باباخانی، ویرایش دوم، تهران، موسسه علمی-فرهنگی نص، ۱۳۸۷.

2.E. Kurniawan & I. Riadi, "Security Level analysis Of academic information systems based on standard ISO 27002:2003 using SSE-CMM", vol. 16, no. 1, pp. 139-147, 2018.

3.I. Riadi, E. I. Aristianto & A. Dahlan, "An Analysis of Vulnerability Web Against Attack Unrestricted Image File Upload", Comput. Eng. Appl., vol. 5, no.1, PP. 19-28, 2016.

4.H. Kim, M. Lee, D. K. Kim, S. K. Chung & K. Chung, "Design and implementation of crypto co-processor and its application to security systems", Berlin Heidelberg: Springer, Computational Intelligence and Security. Lecture Notes in Computer Science, vol. 3802, PP. 1104-9, 2005.

5.S. Reddy, R. Sakthivel & P. Praneet, "VLSI implementation of AES crypto processor for high throughput," In: (IAEST) International Journal of Advanced Engineering Sciences and Technologies, vol. 6, PP. 2-6, 2011.

۶. دری، قیاسیان، سعیدی، «طراحی و پیاده‌سازی رمزنگار AES در بستر FPGA برای خطوط پرسرعت»، مجله مهندسی برق دانشگاه تبریز، جلد ۴۶، شماره ۱، بهار ۱۳۹۵.

7.R. Glabb, L. Imbert, G. Jullien, A. Tisserand and N. Veyrat-Charvillon, "Multi-mode operator for SHA-2 hash functions", JSystArchit 53(2-3):127-38, 2007.

8.N. Sklavos & O. Koufopavlou, "On the hardware implementations of the SHA-2 (256,384,512) hash functions", In Proceeding of the 2003 International

۱۰۲۴	-	۸۱/۰۱	۲۶۶۷	Virtex-4	[۳۷]
۴۵۵۴/۲	-	۱۷۷/۹	۹۸۶	Virtex-6	[۳۸]
۵۰۷۹/۱	-	۱۹۸/۴	۱۰۲۱	Virtex-7	
۱۷۴/۶۰	۳۴۴	۵۸/۶۵۶	۸۴۳	Virtex-4	طرح
۵۱۷/۴۱	۳۴۴	۱۷۳/۸۱۸	۴۳۹	Virtex-5	پیشنهاد
۶۰۶/۴۱	۳۴۴	۲۰۳/۷۱۵	۱۸۱	Virtex-7	ی

با بررسی دقیق جدول‌های ۲ و ۳ و همچنین مطالعه دقیق آنچه که به تفصیل توضیح داده شد؛ می‌توان نتیجه گرفت که با استفاده از معماری پیشنهادی به واسطه بکارگیری برخی راهکارها در راستای به حداقل رساندن ارجاع به حافظه در پروسه پردازشی و استفاده از روش‌های عملکرد موازی، توانسته در ضمن استفاده از حداقل منابع که منجر به کاهش مساحت می‌شود، از سرعت عملیاتی و فرکانس بالایی نسبت به تحقیقاتی با رویکردهای مشابه بهره برد.

#### ۵- نتیجه‌گیری

در این مقاله معماری پیشنهادی پردازنده مبتنی بر FPGA برای الگوریتم‌های رمزنگاری سری SHA-2 معرفی شده است. این معماری به واسطه بکارگیری از واحدهای مسیرداده و حافظه چندپورته، استفاده از ثبات‌های موقت در واحد مسیر داده، استفاده از بانک ثبات به جای حافظه‌های RAM جهت ذخیره داده‌ها در واحد حافظه باعث عملکرد موازی و کاهش دفعات ارجاع به حافظه می‌شود؛ که نتیجه آن‌ها یک معماری متراکم ساده و یکپارچه با سرعت بالا است.

نتایج به دست آمده از پیاده‌سازی طرح پیشنهادی حاکی از آن است که پردازنده متراکم پیشنهادی در مقایسه با کارهای پیشین با اهداف مشابه، توانسته با افزایش فرکانس کاری ۲۵٪ برای SHA-256 و اشغال ۵۵٪ مساحت کمتر برای SHA-512 حد مطلوبی از توان عملیاتی و کارایی را نیز حفظ نماید. پردازنده SHA پیشنهادی برای کاربردهایی مانند بسترهای سیار مورد اعتماد (TMP)، واحد پول دیجیتال (Bitcoin) و مسیریابی ایمن در شبکه‌های روی تراشه‌ها (NoC) مناسب است.

با توجه به مزیت‌های طرح پیشنهادی می‌توان از ساختار ساده معماری پیشنهادی برای طراحی پردازنده جهت اجرای عملیات مربوط به الگوریتم SHA-3 نیز در راستای حفظ



- implementation of acryptographic co-processor”, In: Proceedings: IEEE International Conference on Field-Programmable Technology (IEEE Cat. No.04EX921), PP. 279–285, 2005.
- 17.R. Lien, T. Grembowski and K. Gaj, 1Gbit/s Partially Unrolled Architecture of Hash Function SHA-1 & SHA-512, In: Topics in Cryptography a CT-RSA, pp.1995-1999, 2004.
- 18.A.P. Kakarountas, H. Michail, A. Milidonis, C.E. Goutis & G. Theodoridis, “High-speed FPGA implementation of secure hash algorithm for IPsec and VPN applications”, Journal of Supercomputing, vol. 37, PP.179-195, 2006.
- 19.Y. Yang, F. Chen, Z. Sun, S. Wang, J. Li, J. Chen & Z. Ming, “Secure and efficient parallel hash function construction and its application on cloud audit”, Soft Computing, vol. 23, pp. 8907-8925, 2019.
- 20.Y. K. Lee, H. Chan and I. Verbauwhede, “Iteration bound analysis and throughput optimum architecture of SHA-256(384,512) for hardware implementations”, In: Proceedings of the 8th International Conference on Information Security Applications, vol 256, PP.102–114, 2007.
- 21.M. Kim, J. Ryou and S. Jun, “Efficient hardware architecture of SHA-256 algorithm for trusted mobile computing,” Information Security & Cryptology, Lecture Notes in Computer Science, Berlin Heidelberg: Springer, vol.5487, p. 240–52, 2009.
- 22.R. Chaves, G. Kuzmanor, L. Sousa and S. Vassiliadis, “Cost efficient SHA hardware accelerators”, IEEE Transaction on Very Large Scale Integration Systems, Vol.16, NO.8, PP.999-1008, 2008.
- 23.G. Feng, P. Jain and K. Choi, “Ultra-low power and high speed design and implementation of AES and SHA1 hardware cores in 65 nanometer CMOS technology”, In Electro/Information Symposium on Circuits and Systems, ISCAS’03, vol.5; 2003, p.V-153-V-156, 2003.
- 9.K. Ting, S. Yuen, K. Lee and P. Leong, “An FPGA based SHA -256 processor”, In: Glesner M, Zipf P, Renovell M, editors. Field-Programmable Logic and Applications: Reconfigurable Computing is Going Mainstream, Lecture notes in Computer Science, vol. 2438. Berlin/Heidelberg: Springer, p. 449–71, 2002.
- 10.<http://www.en.wikipedia.org/wiki/securehashalgorithm>.
- 11.U.S. Department of Commerce National Technical Information Service, *FIPS180-2–Secure Hash Standard*, <http://www.csrc.nist.gov/publications/fips/fips180-2/fips180-2>, 2002.
- 12.N. Sklavosand & O.Kou FOPAVLOU, “Implementation of the SHA-2 hash family standard using FPGAs”, Springer Science+Business Media, Inc. Manufactured in the Netherlands, The Journal of supercomputing, vol. 31, pp. 227-248, 2005.
- 13.A. Regenscheid, R. Perlner, S. Chang, J. Kelsey, M. Nandi and S. Paul, Status Report on the First Round of the SHA-3 Cryptographic Hash Complete Competition, NIST, 2009.
- 14.E. Andreeva, B. Mennink and B. Preneel, “Security reductions of the second round SHA\_3 candidates,” In: Proceedings of the 13th International Conference on Information Security, ISC’10. Berlin, Heidelberg: Springer-Verlag, PP. 39–53, 2011.
- 15.P. Kotewar, R. Mandavgane and D. Khatri, “Review on area optimization and simulation of SHA-3,” Journal of Emerging Technologies and Innovative Research (JETIR), PP.290-292, 2014.
- 16.F. Crowe, A. Daly, T. Kerins and W. Marnane, “Single-chip FPGA

efficient SHA-256 hash algorithm for secure vehicle communication using FPGA”, IEEE, ISOC2014, PP. 224-226, 2014.

33.H. Michail, “On the exploitation of a high-throughput SHA-256 FPGA design for HMAC”, ACM Trans on Reconfigurable Tech. and Sys., vol. 5 no. 1, pp. 1- 28, 2012.

34.M.Zeghid, B. Bouallegue, A. Baganne, M. Machhoutand &R. Tourki, “A reconfigurable implementation of the new secure hash algorithm”, Proc Second Int. Conf. Availability, Reliability and Security, (ARES2007), 10–13 April 2007.

35.P. Zawleski, M. Lukowiakand &S. Radziszowsk, *Case Study on FPGA Performance of Parallel Hash Function*, PrzegladElectrotechniczny/ElectricalReview, PP. 151-155, 2010.

36.H. Technology, Efficient Tiny Hash Core Family for Xilinx FPGA Datasheet, Helion Technology Limited, 2010.

37.I. Algreto-Badillo, M. Morales-Sandoval, C. Feregrino-Urbe and R. Cumplido, “Throughput and efficiency analysis of unrolled hardware architectures for the SHA-512 hash algorithm”, IEEE Computer Society Annual Symposium on VLSI, PP. 63- 68, 2012.

38.G. Athanasiou, H. Michail, G. Theodoridis and C. Goutis, “Optimising the SHA-512 cryptographic hash function on FPGAs”, Published in IET Computers & DigitalTechniques, PP.70-83, 2013.

Technology, IEEE International Conference, PP.405-410, 2009.

24.M. Sumagita&I. Riadi, “Analysis of Secure Hash Algorithm (SHA) 512 for Encryption Process on Web Based Application”, IJCSIS Int. J. Dig. Foren. Cyb. Secur., vol. 7, no. 4, pp. 373-381, 2018.

25.M.I.Mazdadi, I. Riadi&A. Luthfi, “Live Forensics on RouterOS API Services to Investigate Network Attacks”, Int. J. Comput. Sci. Inf. Secur., Vol. 15, no. 2, pp.406-410, 2017.

26.W. Stallings, Cryptography and Network Security Principles and Practice, Fifth Edition, Pearson Education, Inc., Publishing as Prentice Hall, 2011.

27.A.L. Barkatullah& T. GailaniCelebi, Design and FPGA Implementation of Hash Processor, Master of Science thesis, Middle East Technical University, 2007.

28.R. Garcia, I. Algreto-Badillo, M. Morales-Sandoval, C. Feregrino-Urbeand and R. Cumplido, “A compact FPGA-based processor for the secure hash algorithm SHA-256”, Elsevier, Computers and Electrical Engineering, vol. 40, pp. 194-202, 2014.

29.H. E. Michail, G.S.Athanasiou, G. Theodoridisand & C. E. Goutis, “On the development of high-throuput and area-efficient multi-mode cryptographic hash designs in FPGA”, Elsevier, Integration, The VLSI Journal, vol. 47, pp.387-407, 2014.

30.X. Cao, L. Luand & M. O’Neill, “A compact SHA-256 architecture for RFID tag”, In: Proceedings of the 22nd IET Irish Signals and Systems Conference, ISSC, Trinity CollegeDublin, 2011.

31.M. Kim, D. Lee and J. Ryou, “Compact and Unified Hardware Architecture for SHA-1 and SHA-256 of Trusted Mobile Computing”, PersUbiquitComput 2012:1–12, <http://www.dx.doi.org/10.1007/s00779-012-0543>.

32.C. Jeong& Y. Kim, “Implementation of

